Um Exemplo de Análise Exploratória dos Dados

Perfil dos Alunos do Curso em 2024

João Pedro Albino

16-09-2024

1. Introdução

Como exemplo introdutório à Ciência de Dados, apresentamos neste trabalho, um roteiro básico para realizar Análise Exploratória de Dados (AED) utilizando a linguagem R (https://cran.r-project.org, a IDE RStudio (https://posit.co/download/rstudio-desktop/) e o *sistema de publicação científica e técnica Quarto* (https://quarto.org), ambos da Posit (https://posit.co).

Será utilizado um *dataset* já **transformado** e **preparado** com a extensão CSV (**comma-separated value** ou seja, **valores separados por vírgulas**), isto é, um arquivo de texto com formato específico para possibilitar o salvamento dos dados em um formato estruturado de tabela. Este formato é amplamente compatível com a linguagem R.

2. Etapas de uma AED

De acordo com Laurett (2011), a finalidade da Análise Exploratória de Dados (AED) é a de examinar previamente os dados antes da aplicação de qualquer técnica estatística. Desta forma o analista consegue, por meio da AED, um entendimento básico de seus dados e das relações existentes entre as variáveis analisadas.

Após a coleta e a estruturação dos dados em uma estrutura dadequada, o próximo passo é a **análise descritiva**. Esta etapa é fundamental, pois uma análise descritiva detalhada permite ao pesquisador familiarizar-se com os dados, organizá-los e sintetizá-los de forma a obter as informações necessárias do **dataset** (conjunto de dados) para responder as questões que estão sendo estudadas.

No exemplo aqui apresentado, serão utilizadas as seguintes etapas:

1. **Carregamento dos Dados**
   - Carregar o arquivo CSV no R.
   - Verificar as primeiras linhas do dataset e a estrutura dos dados.
2. **Limpeza dos Dados**
   - Verificar e tratar valores ausentes.
   - Checar inconsistências e tipos de dados.
3. **Análise Descritiva**
   - Resumo estatístico das variáveis numéricas.
   - Contagem de frequências para variáveis categóricas.
   - Visualização da distribuição das variáveis.
4. **Análise de Correlação**:
   - Calcular a matriz de correlação para variáveis numéricas.
   - Identificar possíveis relações entre variáveis.
5. **Visualização dos Dados**
   - Histogramas, boxplots e scatter plots para entender a distribuição e as relações das variáveis.

6. **Insights e Conclusões**
   - Interpretação dos resultados da análise e geração de insights relevantes.
7. **Documentação com Quarto**
   - Criação de um documento em Quarto que inclui todo o código R, análises e visualizações.
   - Publicação do documento, se necessário.

O script a seguir é um exemplo básico de código em R para realizar a Análise Exploratória de Dados usando um arquivo CSV fornecido e o sistema de publicação Quarto da Posit.

2.1. Carregamento dos Dados

O primeiro **code chunk** (*bloco* ou trecho de código) deve carregar as bibliotecas necessárias para a **sessão** do programa R em execução. Ao executar o R, o usuário inicia uma sessão, a qual persiste até o momento que o usuário fecha o programa ou encerra a sessão (FREIRE, 2021).

Os code chunks, de acordo com Guerra et. al. (2020), são pedaços de código em R que podem ser executados para gerar resultados que serão incorporados ao documento gerado pelo Quarto.

```
# Carregar pacotes necessários
library(tidyverse)
library(readr)
library(summarytools)
library(ggplot2)
library(corrplot)
```

Os dados originais do questionário on-line (https://forms.gle/vNgwk6iLKrgDUiiA9) contém 27 respostas, englobando a totalidade dos alunos matriculados nas disciplinas. Para realizar o tratamento estatístico das informação utilizando o ecossistema R/RStudio, torna-se necessário transformar as respostas digitadas no formulário original em planilhas, por seu formato *tabular*, ou seja, cada indivíduo participante será representado por uma linha na planilha e cada uma de suas resposras em diferentes colunas.

Como as ferramentas que compõem o Google Workspace são integradas, o ambiente permite que as respostas possam ser armazenadas em planilhas de diversos formatos, tais como *Open Document/Open Office* (.ods), arquivos de texto de formato separado-por-vírgula (.csv) e formato de Planilha Excell, da Microsoft (.xlsx) através de comandos internos ao Workspace.

Para esta análise-modelo, numa primeira etapa os dados foram baixados do *Google Forms* em formato de Planilha Excell. Após o download as colunas contendo a data de reposta ao formulário e o endereço de e-mail dos respondentes foram retiradas manualmente, com o objetivo de manter a anonimdade dos participantes.

Além dessa *transformação inicial* a descrição (o nome) das colunas também foram manualmente reduzidas para representar aquela coluna de forma concisa. Além disso, a planilha em Excell foi *salva* em formato *CSV* (Comma-Separated Values) uma estrutura de dados amplamente utilizada em programas de análise de dados devido a sua simplicidade, compatibilidade, tamanho reduzido, flexibilidade e portabilidade. Tais características tornam o formato *CSV* uma escolha popular para a manipulação e análise de dados.

Todas as manipulações relatadas anteriormente foram realizadas com o intuito de **limpar, organizar e consolidar os dados** de forma a deixá-los mais compreensíveis e facilitar o processo de serem importados para o ambiente de análise e na linguagem R.

Dessa forma, o próximo passo da etapa de carregamento de dados é o de realizar a importação dos dados da planilha - do ambiente externo em CSV - para um *formato estruturado* de tal forma que possa ser utilizado nos procesimentos de análise, processamento ou armazenamento.

Portanto, o segundo bloco de código mostra a definição de um *caminho* ou a *localização física* do *dataset* e seu diretório, bem como o comando de *importação* do arquivo para a sessão e sua consequente transformação em **data frame** denominado **dados**.

Deve ser observado que, neste projeto-exemplo foi utilizada uma estrutura de *subdiretórios* com o intuito de manter a organização e facilitar o gerenciamento e o desenvolvimento do projeto. De acordo com Muldoon (2018), Mangini (2018) e Rogonodo (2023), esse tipo de organização garante que os projetos em análise de dados sejam intuitivos, organizados e sustentáveis pois tais projetos implicam em muitos casos em um conjunto de *artefatos de dados*, tais como documentos, arquivos em Excell, dados de sites da web e códigos em R e/ou Python.

```
# Caminho do dadosset
arquivo <- ("../dados/perfil-2024.csv")
# Carregar o arquivo CSV
dados <- read.csv(arquivo, sep = ";") # Excell usa ponto e vírgula como separador quando salva em csv!
```

Um *data frame*, também referenciado como um *quadro de dados*, é a estrutura de dados frequentemente utilizada para armazenar conjunto de dados na forma de tabela (dados tabulares) dentro dos programas em R.

Segundo Tatsch (2021), um quadro de dados é retangular como o formato ou estrutura de uma matriz. Mas apresenta a vantagem de armazenar *vetores* de diferentes tipos (`character`, `numeric`, `logical` e etc) nas suas colunas, se constituindo, portanto, em uma estrutura de armazenamento de dados heterogênea.

Um vetor (ou vector) é uma estrutura ou conjunto de um ou mais elementos *do mesmo tipo*. De acordo com Tatsch (2021), vetor é a estrutura básica de dados na linguagem R e podem ser de dois tipos: vetores atômicos e listas. A Lista é um tipo especial de vetor chamado *list* que é capaz de armazenar dados de diferentes tipos (heterogêneos).

O próximo bloco de código a função **head** retorna ou *lista* as primeiras linhas do dataframe criado, apenas para verificação visual das primeiras **observações.**

```
# Verificar as primeiras linhas do dadosset
head(dados)
```

```
##      genero   tipo_alunoa              formacao             nome_curso
## 1 Masculino Pós-graduação            Tecnologia     Midia e Tecnologia
## 2  Feminino Pós-graduação            Tecnologia     Midia e Tecnologia
## 3 Masculino Pós-graduação     Ciências Contábeis     Midia e Tecnologia
## 4 Masculino Pós-graduação  Ciência da Computação     Midia e Tecnologia
## 5  Feminino Pós-graduação          Artes Visuais     Midia e Tecnologia
## 6 Masculino     Graduação Ensino Médio e Técnico Sistemas de Informação
##         faixa_etaria comp       tipo_comp sis_op_comp fabrica_comp lingua_dados
## 1 Entre 25 e 29 anos  Sim          Laptop       Outro   Multilaser          Sim
## 2 Entre 25 e 29 anos  Sim          Laptop     Windows           HP          Sim
## 3 Entre 30 e 34 anos  Sim          Laptop     Windows         Acer          Não
## 4 Entre 40 e 44 anos  Sim Desktop, Laptop     Windows       Lenovo          Não
## 5 Entre 22 e 24 anos  Sim          Laptop     Windows         Dell          Não
## 6 Entre 18 e 21 anos  Sim          Laptop     Windows         Acer          Sim
##   lingua_prog
## 1      Python
## 2 Linguagem R
## 3     Nenhuma
## 4     Nenhuma
## 5     Nenhuma
## 6      Python
##
## 1
## 2
```

```
## 3 Proporcionar um entendimento de conceitos e técnicas fundamentais, para análise de grandes volumes
## 4
## 5
## 6
```

Já o próximo bloco, utizando a função **str**, exibe (mostra) de forma compacta a estrutura de qualquer objeto na linguagem R, no caso, exibe a estrutura do dataframe *dados*.

```
# Verificar a estrutura dos dados
str(dados)
```

```
## 'data.frame':    27 obs. of  12 variables:
##  $ genero      : chr  "Masculino" "Feminino" "Masculino" "Masculino" ...
##  $ tipo_alunoa : chr  "Pós-graduação" "Pós-graduação" "Pós-graduação" "Pós-graduação" ...
##  $ formacao    : chr  "Tecnologia" "Tecnologia" "Ciências Contábeis" "Ciência da Computação" ...
##  $ nome_curso  : chr  "Midia e Tecnologia" "Midia e Tecnologia" "Midia e Tecnologia" "Midia e Tecnol
##  $ faixa_etaria: chr  "Entre 25 e 29 anos" "Entre 25 e 29 anos" "Entre 30 e 34 anos" "Entre 40 e 44 a
##  $ comp        : chr  "Sim" "Sim" "Sim" "Sim" ...
##  $ tipo_comp   : chr  "Laptop" "Laptop" "Laptop" "Desktop, Laptop" ...
##  $ sis_op_comp : chr  "Outro" "Windows" "Windows" "Windows" ...
##  $ fabrica_comp: chr  "Multilaser" "HP" "Acer" "Lenovo" ...
##  $ lingua_dados: chr  "Sim" "Sim" "Não" "Não" ...
##  $ lingua_prog : chr  "Python" "Linguagem R" "Nenhuma" "Nenhuma" ...
##  $ expectativa : chr  "Espero aprender a base sobre o assunto de 'Ciência de Dados', os primeiros pas
```

Observa-se que o código anterior motrou que todas as variáveis do *dataframe* **dados** contém em sua estrutura variáveis do tipo **chr** ou seja, caracter. Entretanto, utilizar dados do tipo **character** (chr) em R para gráficos e análises pode apresentar algumas dificuldades, principalmente porque a linguagem R não trata automaticamente esses dados como categorias distintas.

Para evitar esses problemas, é recomendável converter dados tipo character em **factors** antes de realizar análises ou criar gráficos. Em R, o formato **factor** é utilizado principalmente para representar *dados categóricos*, ou seja, **dados que podem ser divididos em categorias distintas** (SOUZA, 2023).

O bloco de código a seguit mostra como isso foi realizado no dataframe dados.

```
# Converter todas as variáveis do tipo character em fatores, exceto a variável 'expectativa'
dados <- dados %>%
  mutate(across(.cols = where(is.character) & !c("expectativa"), as.factor))
```

Após a execução do bloco de código anterior, podemos verificar que as variáveis do tipo caracter foram todas *transformadas* em * factor*. A variável **expectativa** foi mantida com caracter en função da análise de conteúdo que deverá ser realizada.

```
# Verificar a nova estrutura dos dados
str(dados)
```

```
## 'data.frame':    27 obs. of  12 variables:
##  $ genero      : Factor w/ 2 levels "Feminino","Masculino": 2 1 2 2 1 2 2 2 2 2 ...
##  $ tipo_alunoa : Factor w/ 3 levels "Graduação","Outro",..: 3 3 3 3 3 1 3 3 3 3 ...
##  $ formacao    : Factor w/ 14 levels "Administração ",..: 14 14 4 3 2 9 5 14 14 14 ...
##  $ nome_curso  : Factor w/ 3 levels "Ciência da Computação",..: 2 2 2 2 2 3 2 2 2 2 ...
##  $ faixa_etaria: Factor w/ 9 levels "Acima de 59 anos",..: 4 4 5 7 3 2 3 4 6 9 ...
```

```
##  $ comp        : Factor w/ 2 levels "Não","Sim": 2 2 2 2 2 2 2 2 2 2 ...
##  $ tipo_comp   : Factor w/ 4 levels "Desktop","Desktop, Laptop",..: 3 3 3 2 3 3 3 2 3 2 ...
##  $ sis_op_comp : Factor w/ 3 levels "macOS","Outro",..: 2 3 3 3 3 3 3 3 3 3 ...
##  $ fabrica_comp: Factor w/ 9 levels "Acer","Apple",..: 8 6 1 7 5 1 1 4 1 5 ...
##  $ lingua_dados: Factor w/ 2 levels "Não","Sim": 2 2 1 1 1 2 1 1 1 1 ...
##  $ lingua_prog : Factor w/ 4 levels "Linguagem R",..: 4 1 3 3 3 4 3 3 3 3 ...
##  $ expectativa : chr  "Espero aprender a base sobre o assunto de 'Ciência de Dados', os primeiros pas
```

Os blocos anteriores representam a etapa inicial de muitos programas em linguagem R que utilizam arquivos em sua execução. Em um processo de análise de dados esses processos ocorrem frequentemente.

2.2. Limpeza dos Dados

A etapa de *Limpeza de Dados*, também definida como *data cleaning*, tem o objetivo de *verificar* e *tratar* os valores ausentes (também denominados de **missing values**) no dataframe e também constatar se existem inconsistências e observar os tipos de dados que foram importados.

Os blocos de programa mostrado a seguir, utilizam a função genérica **summary(),** utilizada para produzir resumos dos resultados de várias funções e objetos. Segundo Oliveira (2021), a função *summary* é muito utilizada no processo de estatística descritiva na linguagem R, pois com ela é possível obter a amplitude, a média e a mediana dos dados, porém não a função mão mede o tamanho amostral total, o tipo de distribição dos dados e nem a medida de disperção desses dados.

Entretanto, afirma Oliveira (2021), a função summary() é bastante utilizada por estatísticos pois, ao invés de se obter a descrição estatística dos dados manualmente, a partir de funções individuais, tais como `mean()`, `median()`, `sd()` ou `se()`, `min()`, `max()`, tal forma de descrição dos dados se torna obsoleta ao momento que a função *summary()* representa a forma muito mais simples e rápida de se obter a descrição do que se deseja.

O primeiro trecho de código a seguir, efetua a verificação se existem valores ausentes. Dados ausentes, do inglês, *missing values*, são definidos como **dados que não estão presentes para algumas variáveis no conjunto de dados fornecido** (RODRIGUES, 2023).

O código busca quantificar, de forma mais precisa, o quanto a integridade do dataset está comprometida por valores nulos.

```r
# Verificar se há valores ausentes
summary(is.na(dados))
```

```
##    genero        tipo_alunoa      formacao       nome_curso
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:27        FALSE:27        FALSE:27        FALSE:27
##  faixa_etaria       comp         tipo_comp       sis_op_comp
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:27        FALSE:27        FALSE:27        FALSE:27
##  fabrica_comp    lingua_dados    lingua_prog     expectativa
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:27        FALSE:27        FALSE:27        FALSE:27
```

A parte is.na(dados) do código detecta valores ausentes (se houver), retornando um valor booleano para cada elemento no dataframe.

2.3. Análise Descritiva

Segundo Oliveira (2021), A estatística descritiva é uma parte da estatística que descreve os principais parâmetros e conformação dos dados. A partir disso pode-se tomar conclusões para exploração de dados e para futuras análises que podem vir a ser utilizadas.

Os principais parâmetros dos dados são: o tamanho amostral (n); as medidas de tendência central (média, moda e mediana); medidas de dispersão (variância e desvio padrão); a amplitude dos dados (menor e maior número); e à normalidade dos dados (distribuição gaussiana com a média e mediana sendo a medida central) (OLIVEIRA, 2021).

Geralmente a análises descritivas são as primeiras manipulações realizadas em um estudo quantitativo e tem como principal objetivo resumir, sumarizar e explorar o comportamento dos dados, afirma Previdelli (2017). Isso pode ser feito através de tabelas de frequências, gráficos e medidas de resumo numérico.

O código a seguir objetiva exibir o reumo estatístico das variáveis numéricas do dataframe **dados**.

```
# Resumo estatístico das variáveis numéricas
summary(select_if(dados, is.numeric))
```

```
## < table of extent 0 x 0 >
```

Como pode ser observado, não existem dados numéricos no dataframe, somente dados do tipo caracter e fatores (factor), como pode ser visto na execução da função **str()** no tópico 2.1, que exibiu a estrutura do dataframe *dados*.

Dado que todos as variáveis do dataset **dados** são do tipo **Factor,** faz-se necessária um outro tipo de resumo estatístico, com a verificação, na função summary(), deste formato de dado.

O termo fator se refere a um tipo de dado estatístico usado para armazenar **variáveis categóricas**. A **diferença** entre uma *variável categórica* e uma *variável contínua* é que, uma variável categórica pode pertencer a um **número limitado de categorias**. Uma variável contínua, por sua vez, pode corresponder a um **número infinito de valores** (TATSCH, 2021).

```
# Contagem de frequências para variáveis categóricas
summary(select_if(dados, is.factor))
```

```
##        genero          tipo_alunoa                    formacao
##  Feminino : 7   Graduação    :10   Ensino Médio           :6
##  Masculino:20   Outro        : 2   Tecnologia             :5
##                 Pós-graduação:15   Ensino Médio e Técnico:3
##                                    Administração          :2
##                                    Ciência da Computação  :2
##                                    Artes Visuais          :1
##                                    (Other)                :8
##                    nome_curso             faixa_etaria  comp
##  Ciência da Computação : 2   Entre 18 e 21 anos:8     Não: 1
##  Midia e Tecnologia    :16   Entre 22 e 24 anos:4     Sim:26
##  Sistemas de Informação: 9   Entre 25 e 29 anos:4
##                              Entre 30 e 34 anos:3
##                              Entre 40 e 44 anos:3
##                              Entre 50 e 59 anos:2
##                              (Other)           :3
##                  tipo_comp   sis_op_comp  fabrica_comp lingua_dados
##  Desktop              : 1   macOS  : 1   Acer   :9     Não:12
```

```
##  Desktop, Laptop      : 7   Outro  : 1   Lenovo :7     Sim:15
##  Laptop               :18   Windows:25   Dell   :5
##  Não possuo computador: 1                Apple  :1
##                                          Asus   :1
##                                          Avell  :1
##                                          (Other):3
##                lingua_prog
##  Linguagem R         : 2
##  Machine Learning, IA: 1
##  Nenhuma             :12
##  Python              :12
##
##
##
```

Como pode ser visto nos resultados de exibição do trecho de código, um bom exemplo de variável categórica é a variável `genero`, a qual possui, de forma simplificada e com os dados deste dataset exemplo, dois valores possíveis, e cada observação (cada linha do dataframe) pode ser armazenada com o valor "Feminino" ou "Masculino".

O tipo de de dado *factor* facilita a *plotagem* das variáveis em um gráfico, por exemplo, pois internamente, cada *level* de um fator é representado como um inteiro. No exemplo anterior, se observamos a estrutura da variável `genero`, o level `Feminino` está sendo representado como `1` e o level `Masculino` como `2` (DAMIANI, 2022).

```
str(dados$genero) # Observar o tipo e a estrutura da variável genero
```

```
##  Factor w/ 2 levels "Feminino","Masculino": 2 1 2 2 1 2 2 2 2 2 ...
```

```
dados$genero # Verificar o conteúdo da variável genero
```

```
##  [1] Masculino Feminino  Masculino Masculino Feminino  Masculino Masculino
##  [8] Masculino Masculino Masculino Masculino Masculino Masculino Masculino
## [15] Masculino Masculino Masculino Masculino Feminino  Feminino  Masculino
## [22] Masculino Feminino  Feminino  Feminino  Masculino Masculino
## Levels: Feminino Masculino
```

O bloco de códgo a seguir tem o objetivo de identificar quais são as variáveis categóricas e armazenar os nome destas como um **dataframe** em `categorical_vars`.

```
# Identificar as variáveis categóricas (fatores ou do tipo character)
categorical_vars <- dados %>% select(where(is.factor)) # Neste exemplo, dataframe

head(categorical_vars, 4) # visualizando o conteúdo da variável. Apenas as linhas iniciais
```

```
##      genero   tipo_alunoa             formacao         nome_curso
## 1 Masculino Pós-graduação           Tecnologia Midia e Tecnologia
## 2  Feminino Pós-graduação           Tecnologia Midia e Tecnologia
## 3 Masculino Pós-graduação    Ciências Contábeis Midia e Tecnologia
## 4 Masculino Pós-graduação Ciência da Computação Midia e Tecnologia
##         faixa_etaria comp       tipo_comp sis_op_comp fabrica_comp lingua_dados
## 1 Entre 25 e 29 anos  Sim          Laptop       Outro   Multilaser          Sim
```

```
## 2 Entre 25 e 29 anos  Sim          Laptop     Windows           HP          Sim
## 3 Entre 30 e 34 anos  Sim          Laptop     Windows         Acer          Não
## 4 Entre 40 e 44 anos  Sim Desktop, Laptop     Windows       Lenovo          Não
##   lingua_prog
## 1      Python
## 2 Linguagem R
## 3     Nenhuma
## 4     Nenhuma
```

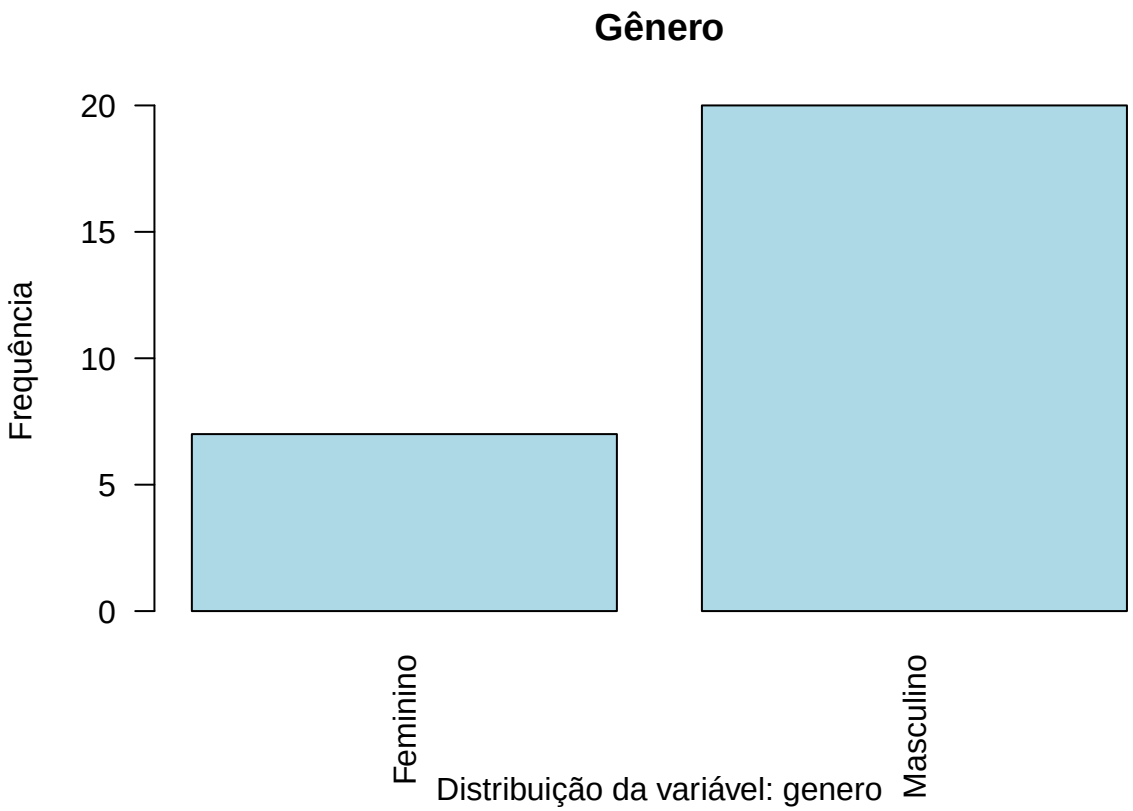
3. Visualização dos Dados

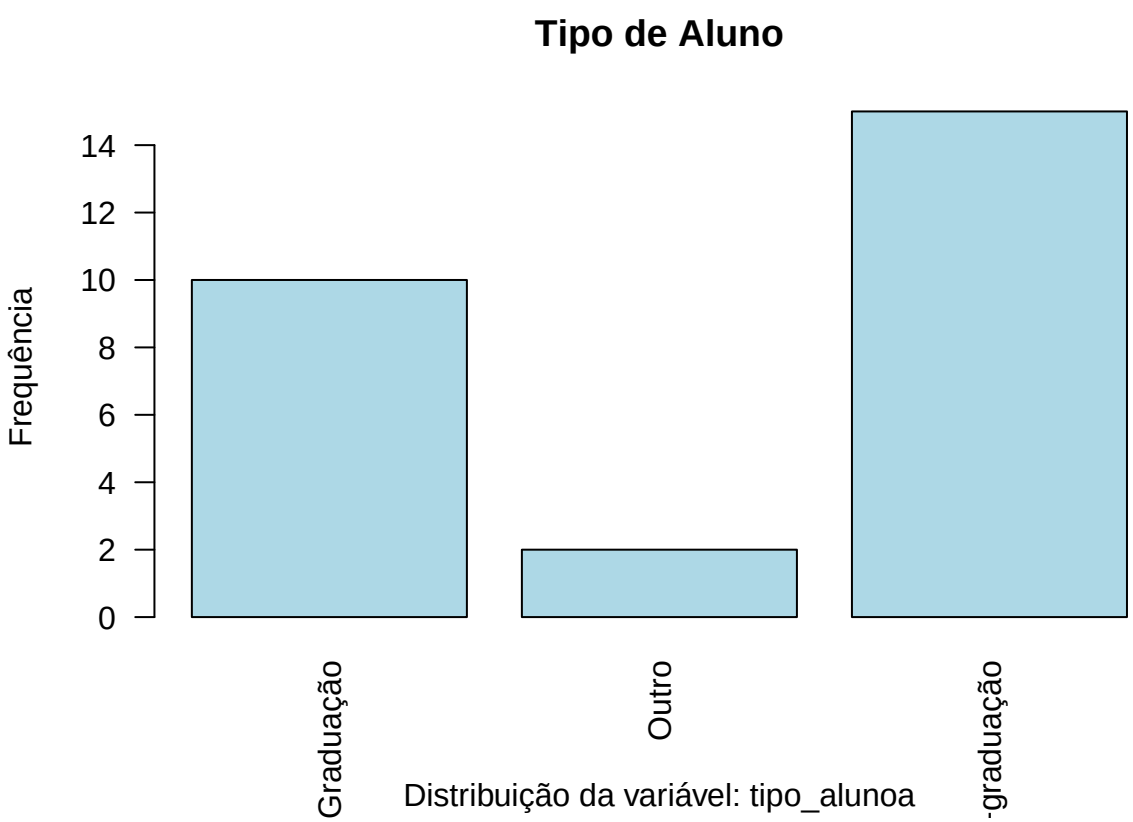
A visualização de gráficos envolve a criação de representações visuais de dados, como *gráficos de barras*, *gráficos de linhas*, *gráficos de dispersão*, *histogramas*, dentre outros. Esses gráficos ajudam a comunicar informações de forma clara e eficiente, permitindo que os espectadores compreendam rapidamente os dados apresentados.

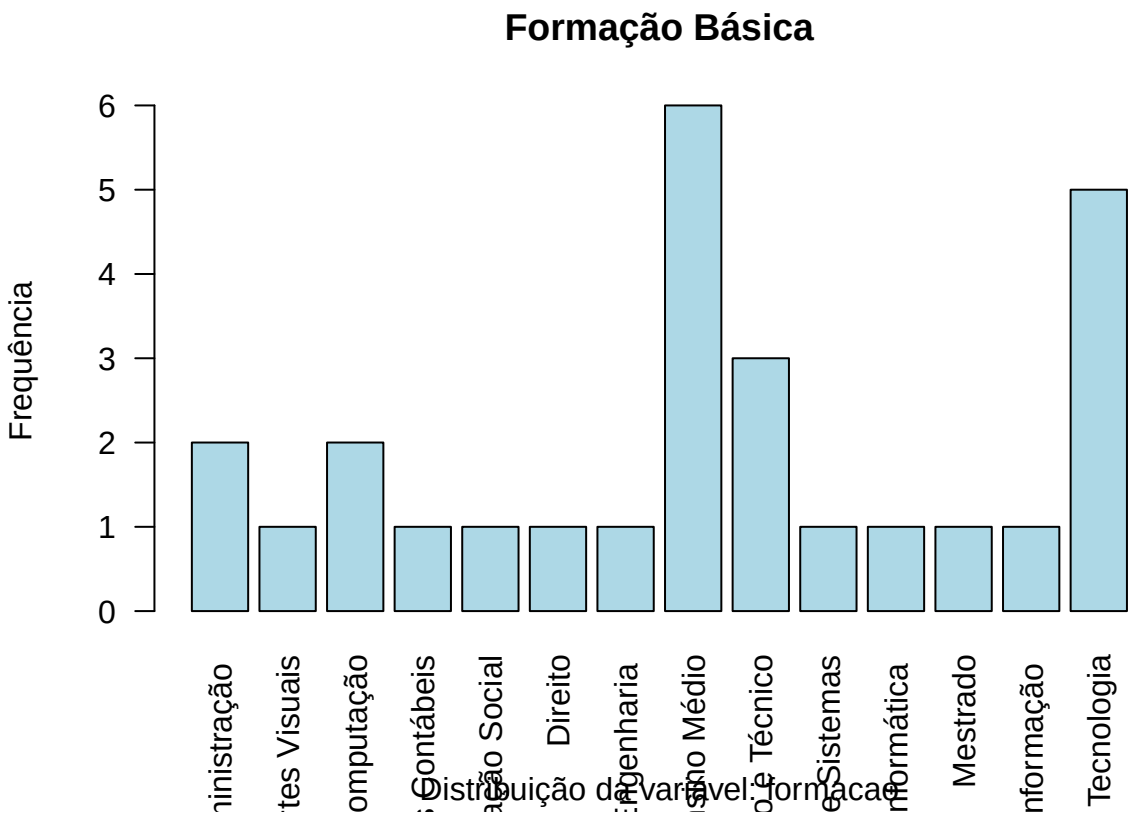
Uma forma fundamental de criar gráficos básicos de visualização de dados na linguagem é R utilizando a função **`plot()`**. A função **`plot()`** é bastante versátil e pode ser usada para criar diferentes tipos de gráficos, como gráficos de dispersão, gráficos de linhas, histogramas, entre outros.

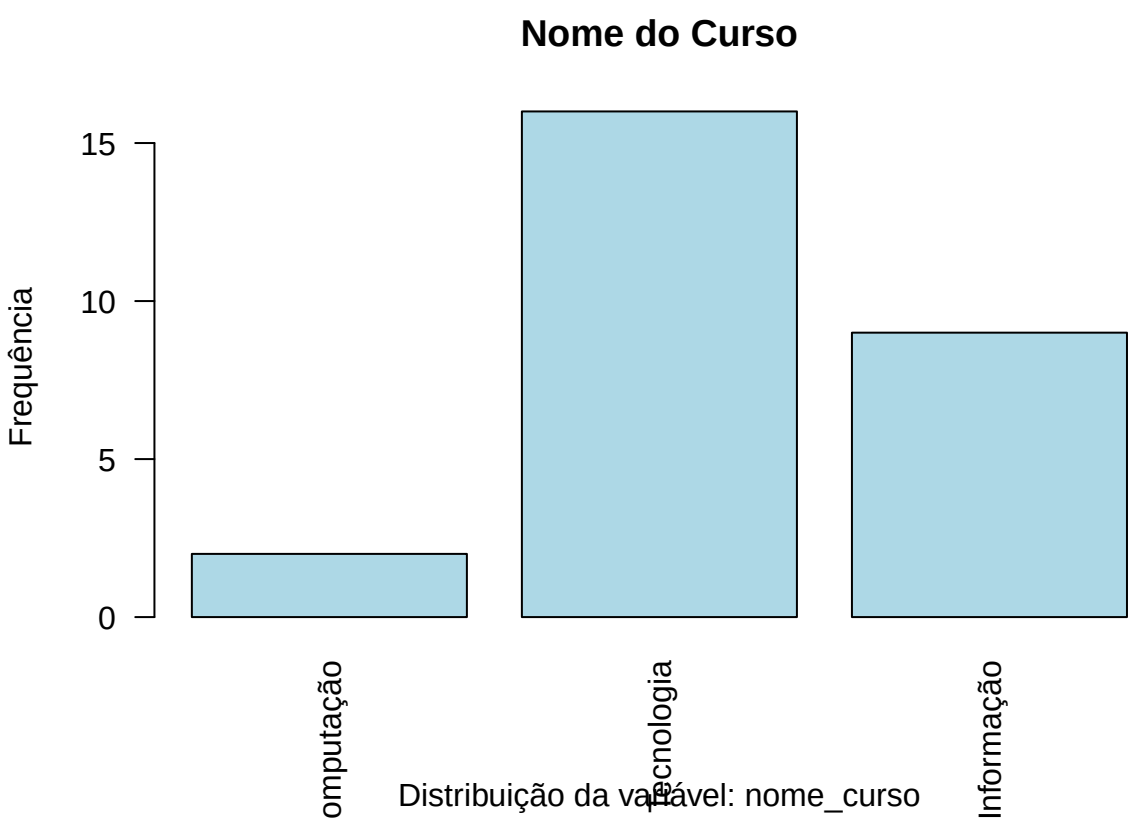
O bloco a seguir mostra a *distribuição das variáveis categóricas* armazenadas no vetor `categorical_vars` através da plotagem dos gráficos de barra utilizando a função **`plot()`**.

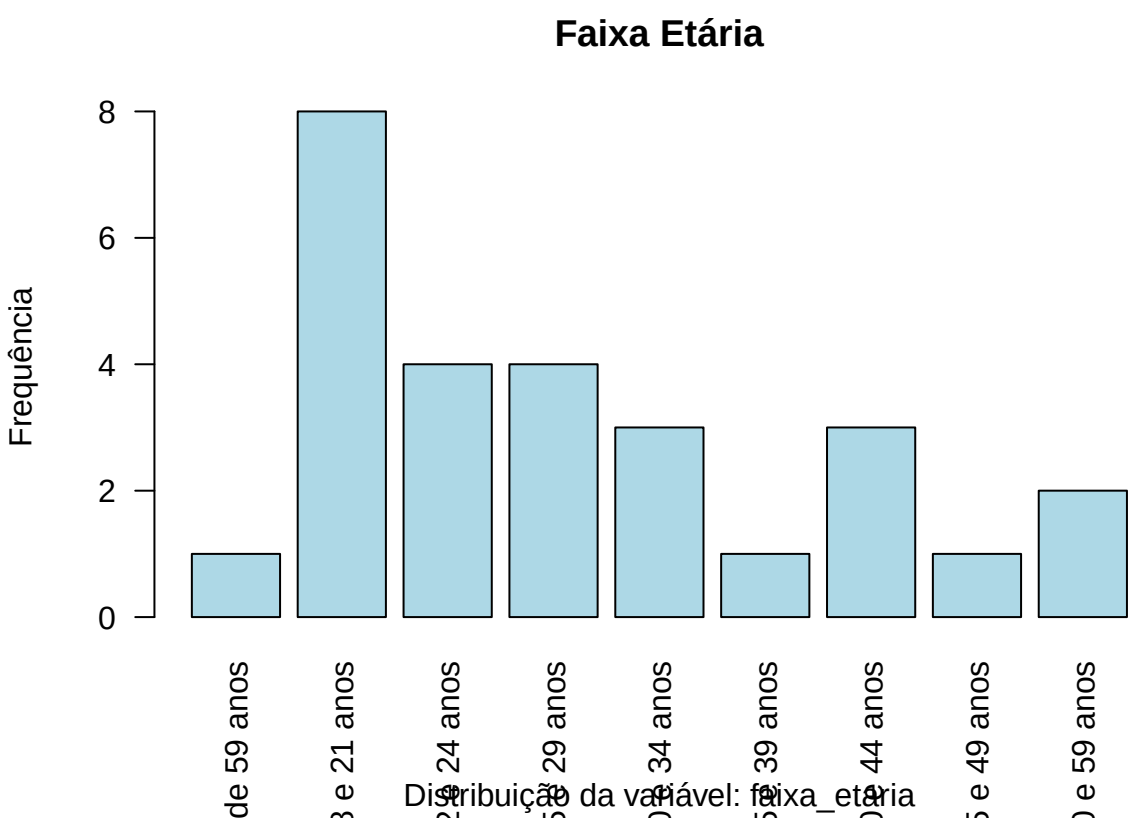
```r
# Atribuindo um nome mais descritivo para as variáveis no título
titulo_grafico = c("Gênero", "Tipo de Aluno", "Formação Básica", "Nome do Curso", "Faixa Etária", "Possu
                   "Tipo do Computador", "Sistema Operaciona Utilizado","Fabricante do Computador",
                   "Conhecimento em Ciência de Dados", "Qual Linguagem Conhece")
indice = 1
for (var_name in names(categorical_vars)) {
  # Selecionar a variável atual
  var_data <- categorical_vars[[var_name]]

  # Gerar o gráfico de barras
  plot(var_data,
       main = titulo_grafico[indice],                          # Título do gráfico
       sub = paste("Distribuição da variável:", var_name),  # Subtítulo do gráfico
#       xlab <- var_name,
       ylab = "Frequência",    # Rótulo do eixo Y
       col = "lightblue",      # Cor das barras
       las = 2)                # Rotacionar os rótulos do eixo X para facilitar a leitura
  indice = indice + 1
}
```

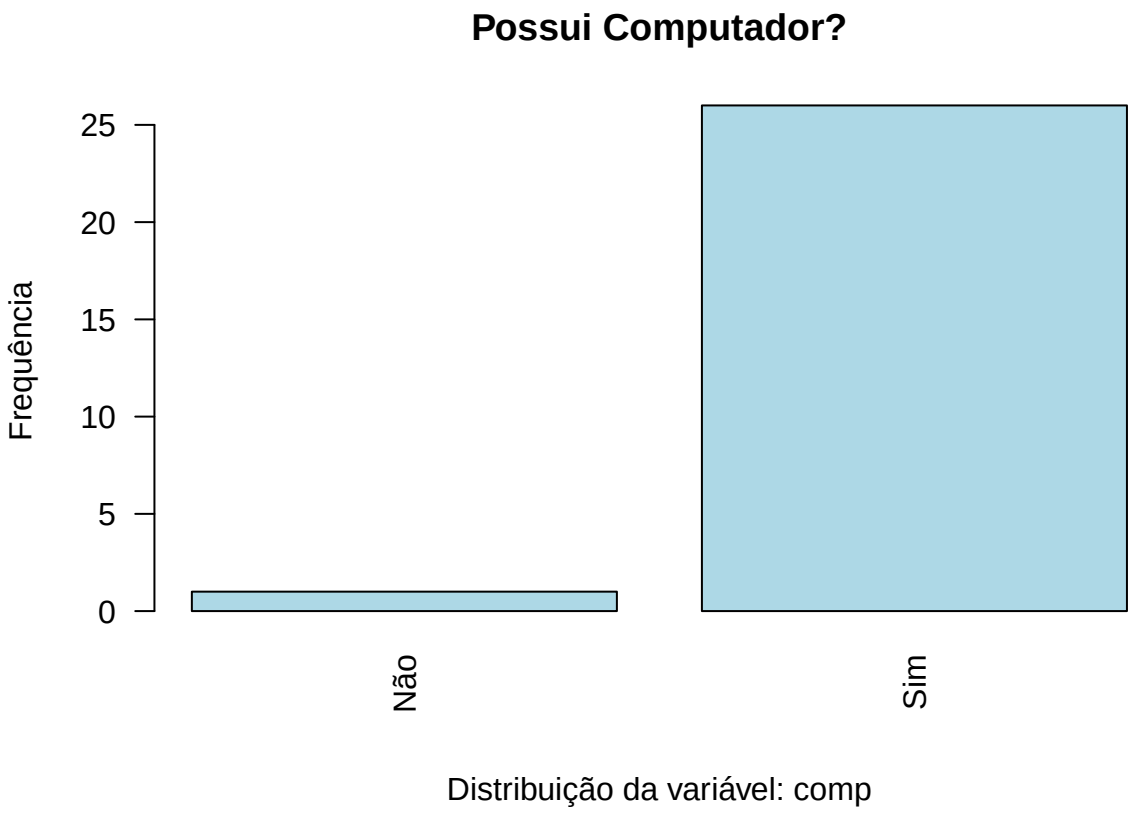

Gênero
20
15
10
5
0
Frequência
Feminino
Masculino
Distribuição da variável: genero

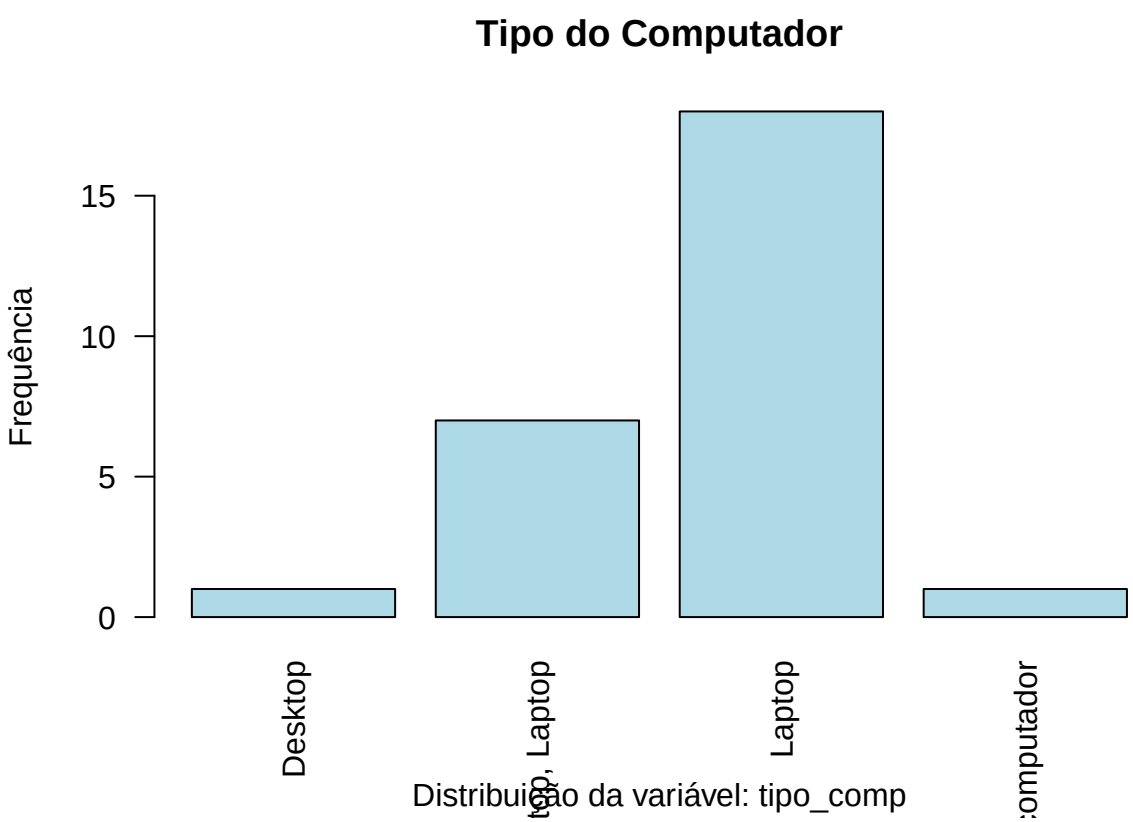

Tipo de Aluno
Frequência
0
2
4
6
8
10
12
14
Graduação
Outro
-graduação
Distribuição da variável: tipo_alunoa

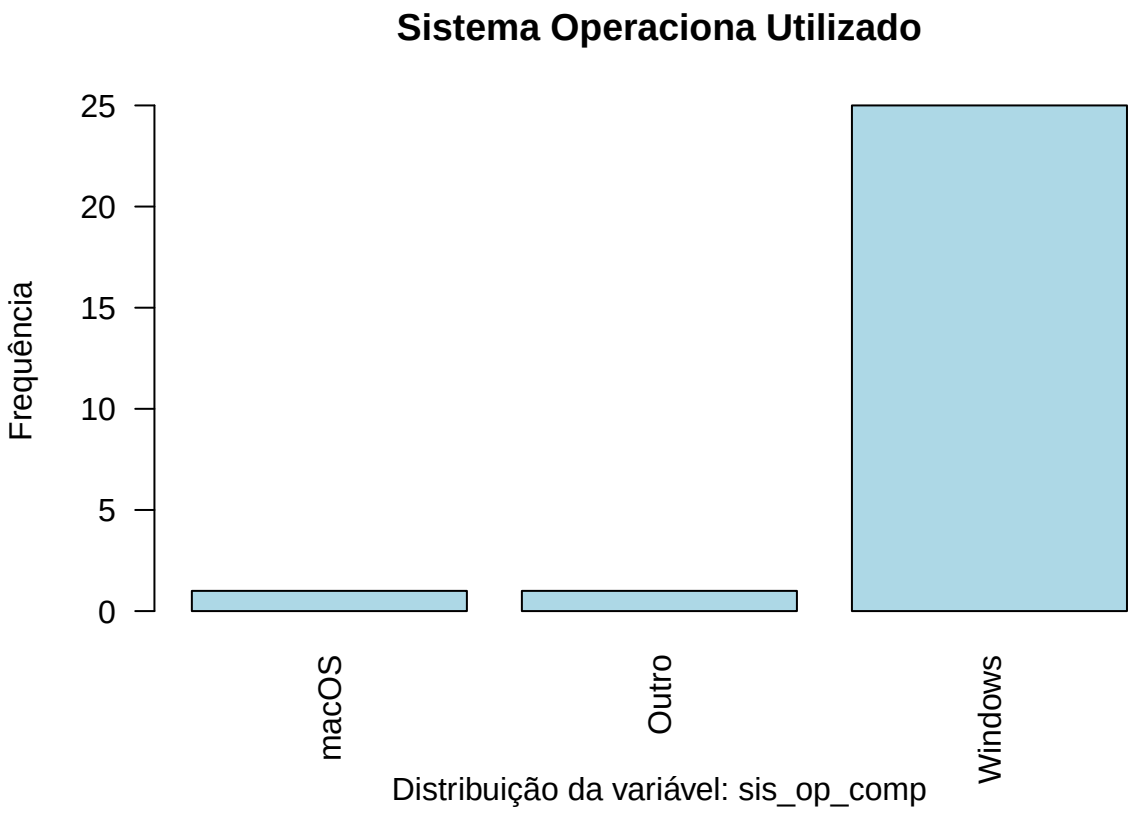

Formação Básica
Frequência
0
1
2
3
4
5
6
Distribuição da variável formacae
Tecnologia
Mestrado
Sistemas
Técnico
Médio
Direito
Social
Contábeis
Visuais
Administração

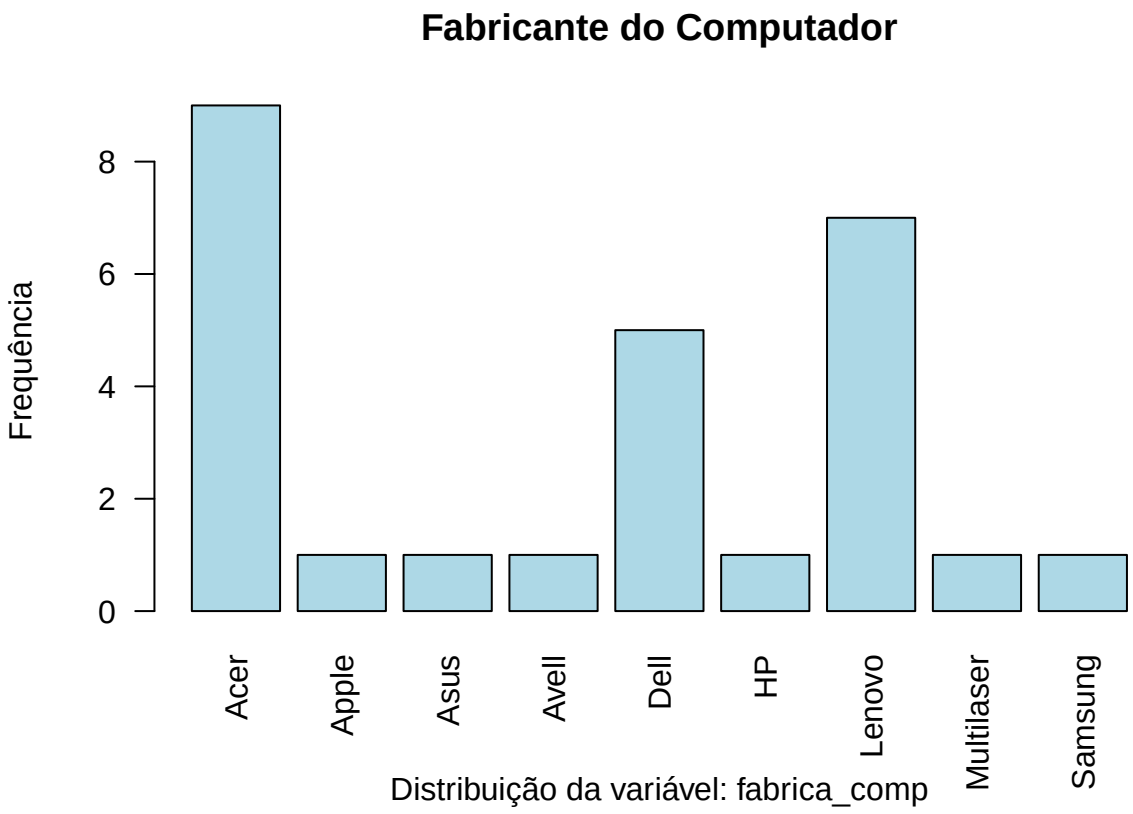

Nome do Curso
Frequência
15
10
5
0
omputação
cnologia
Informação
Distribuição da variável: nome_curso

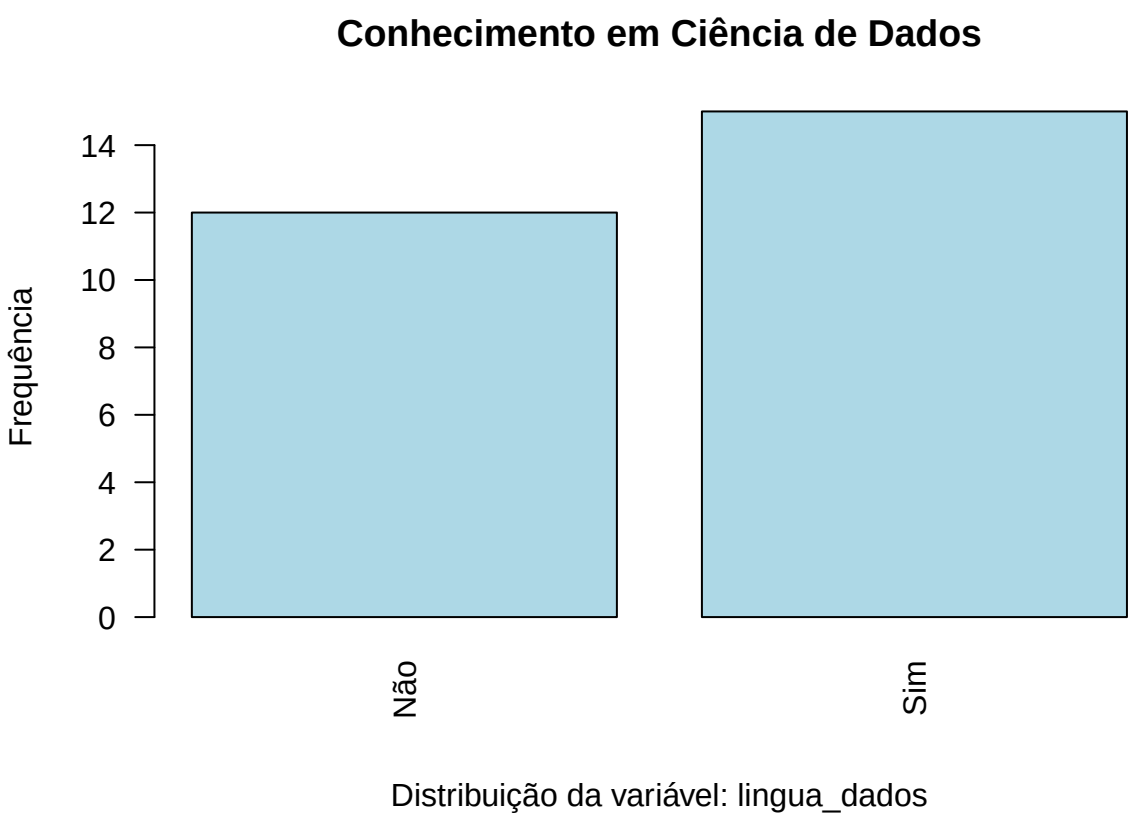

Faixa Etária
Frequência
8
6
4
2
0
de 59 anos
e 21 anos
24 anos
29 anos
34 anos
39 anos
44 anos
e 49 anos
e 59 anos
Distribuição da variável: faixa_etaria

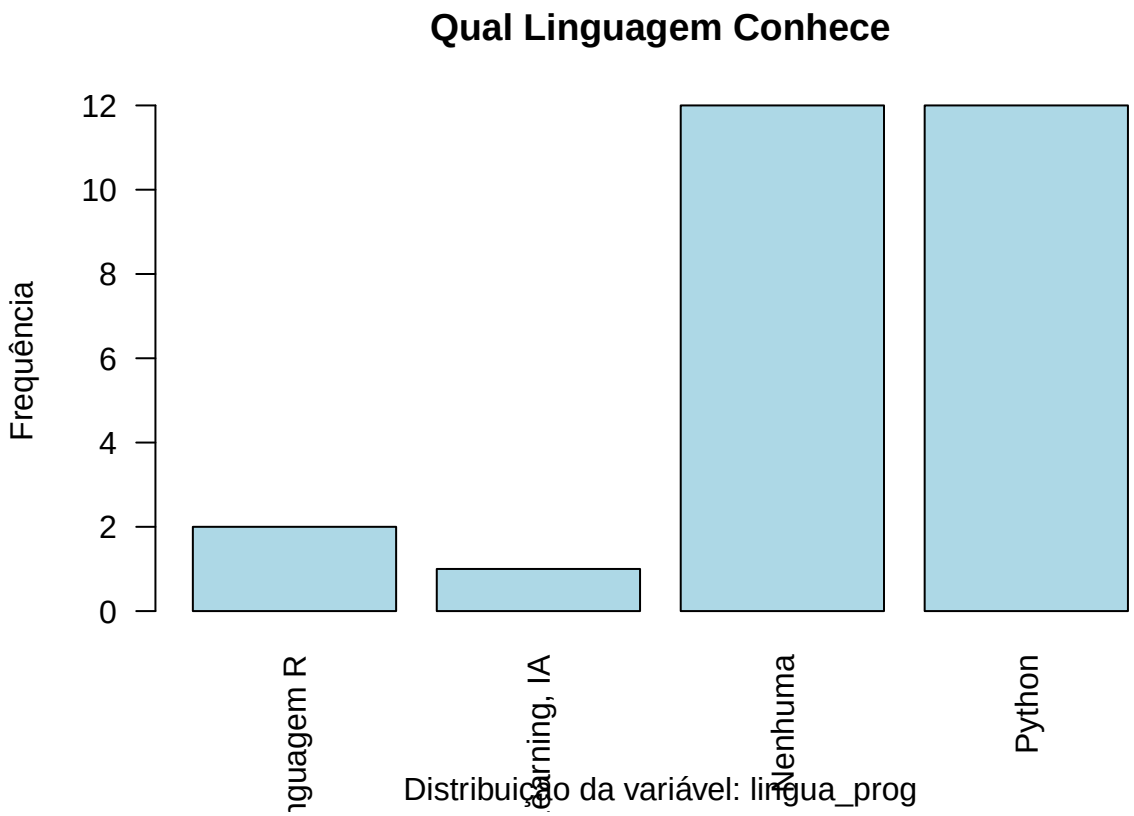

Possui Computador?
Frequência
25
20
15
10
5
0
Não
Sim
Distribuição da variável: comp


Tipo do Computador
Frequência
15
10
5
0
Desktop
top, Laptop
Laptop
computador
Distribuição da variável: tipo_comp


Sistema Operaciona Utilizado
Frequência
25
20
15
10
5
0
macOS
Outro
Windows
Distribuição da variável: sis_op_comp


Fabricante do Computador
Frequência
0
2
4
6
8
Acer
Apple
Asus
Avell
Dell
HP
Lenovo
Multilaser
Samsung
Distribuição da variável: fabrica_comp


Conhecimento em Ciência de Dados
Frequência
14
12
10
8
6
4
2
0
Não
Sim
Distribuição da variável: lingua_dados



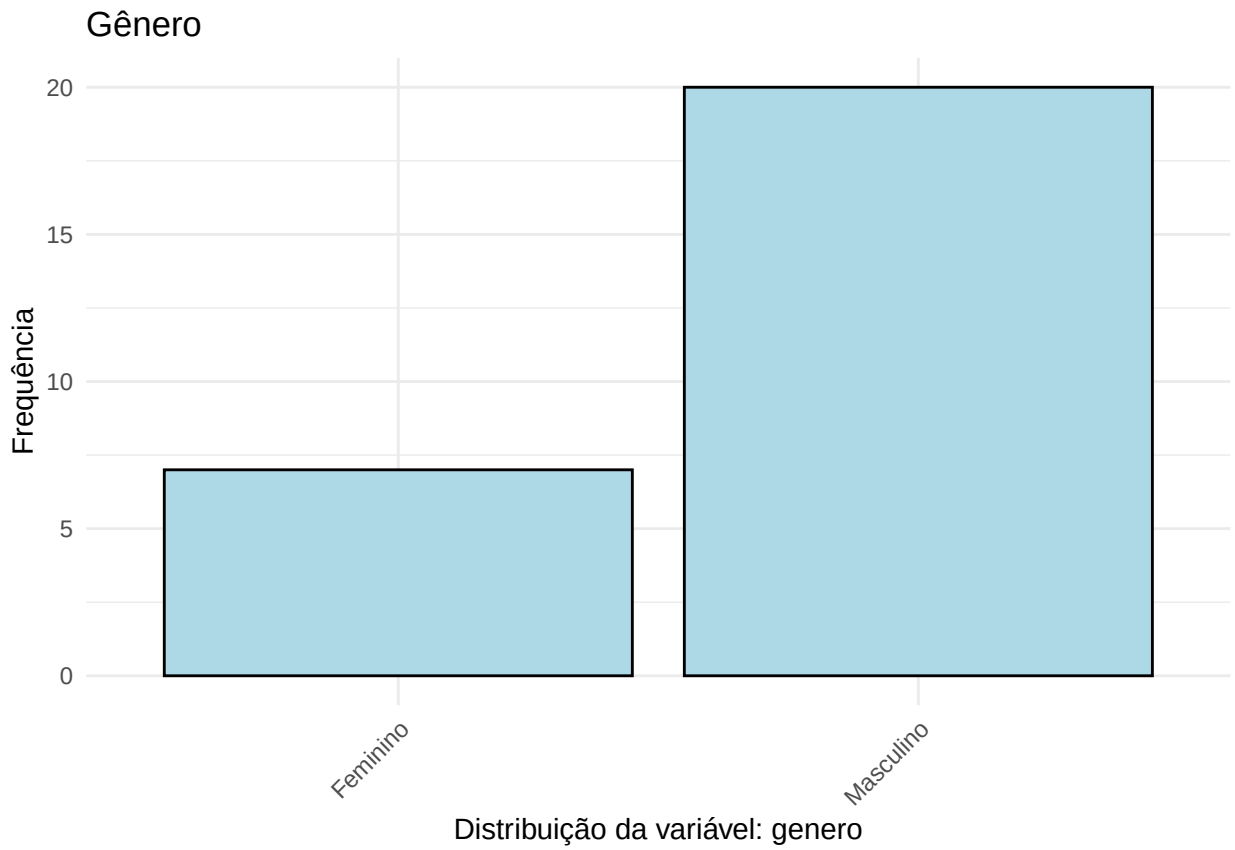
Outra alternativa para a visualização dos dados pode ser realizada por meio da biblioteca **ggplot2()**. O **ggplot2** é um pacote bastante poderoso e flexível do R utilizado para criar visualizações de dados mais elegantes. Essa biblioteca é baseada na **Gramática dos Gráficos**, o que significa que você pode expressar seus gráficos usando um conjunto de elementos combináveis ao invez de estar limitado a tipos de gráficos predefinidos (SARDAR, 2024).

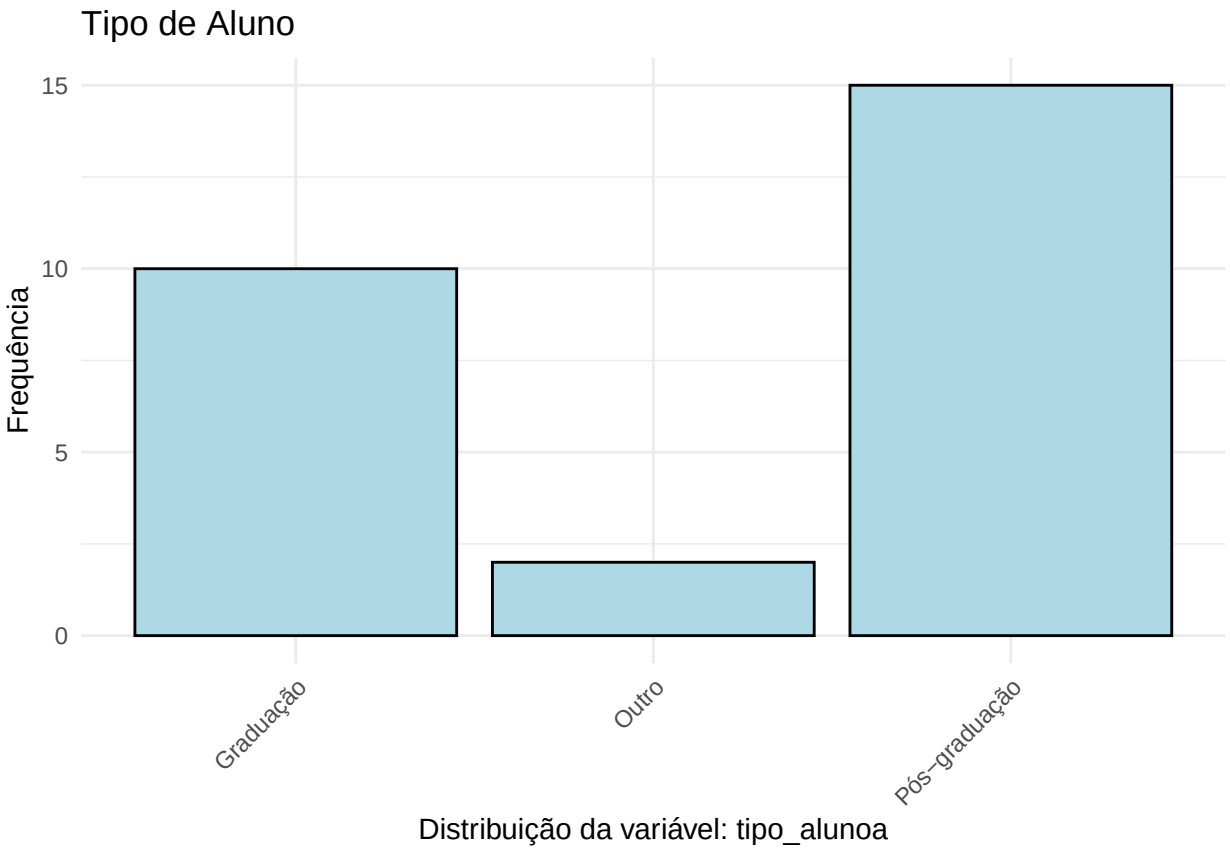
De acordo com Wilkinson (1999), a **Gramática dos Gráficos** é uma abordagem teórica para a criação de gráficos estatísticos. Esta gramática fornece uma estrutura unificada para representar uma ampla variedade de gráficos, permitindo que os usuários construam visualizações complexas de maneira sistemática e coerente. Essa gramática obteve grande destaque com a criação do pacote **ggplot2** por Wickham (2016), o qual implementa essa abordagem na linguagem de programação R. Wickham (2010) também publicou um artigo onde descreveu atualizações e melhorias para a gramática original.

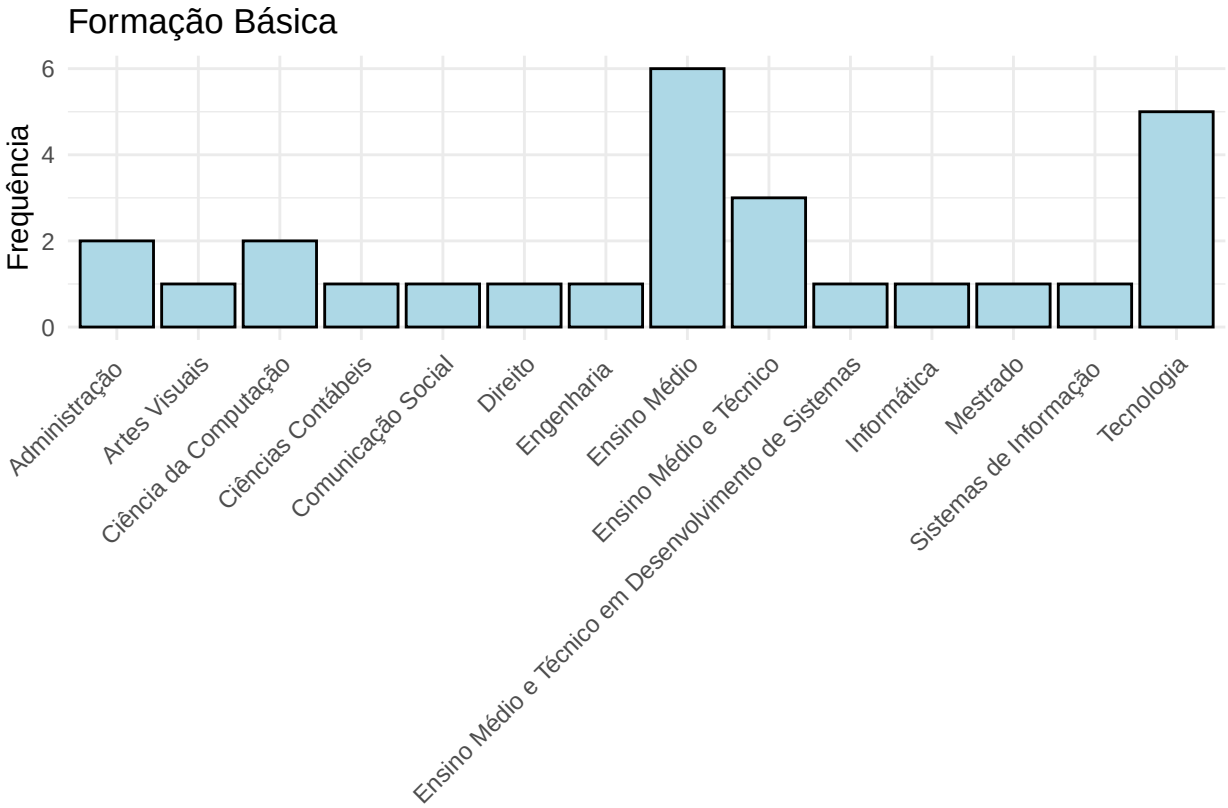
O bloco a seguir mostra a distribuição das variáveis categóricas armazenadas no vetor `categorical_vars` através da plotagem dos gráficos de barra.

```
# Identificar as variáveis categóricas (fatores ou do tipo character)
categorical_vars <- dados %>% select(where(is.factor)) # Aqui "categorical_vars2" é um vetor!

indice = 1
# Gerar gráficos de barras para cada variável categórica
for (var_name in names(categorical_vars)) {
  # Gerar o gráfico de barras usando ggplot2
  p <- ggplot(dados, aes_string(x = var_name)) +
    geom_bar(fill = "lightblue", color = "black") +
    labs(title =  titulo_grafico[indice],
         x = paste("Distribuição da variável:", var_name),
```

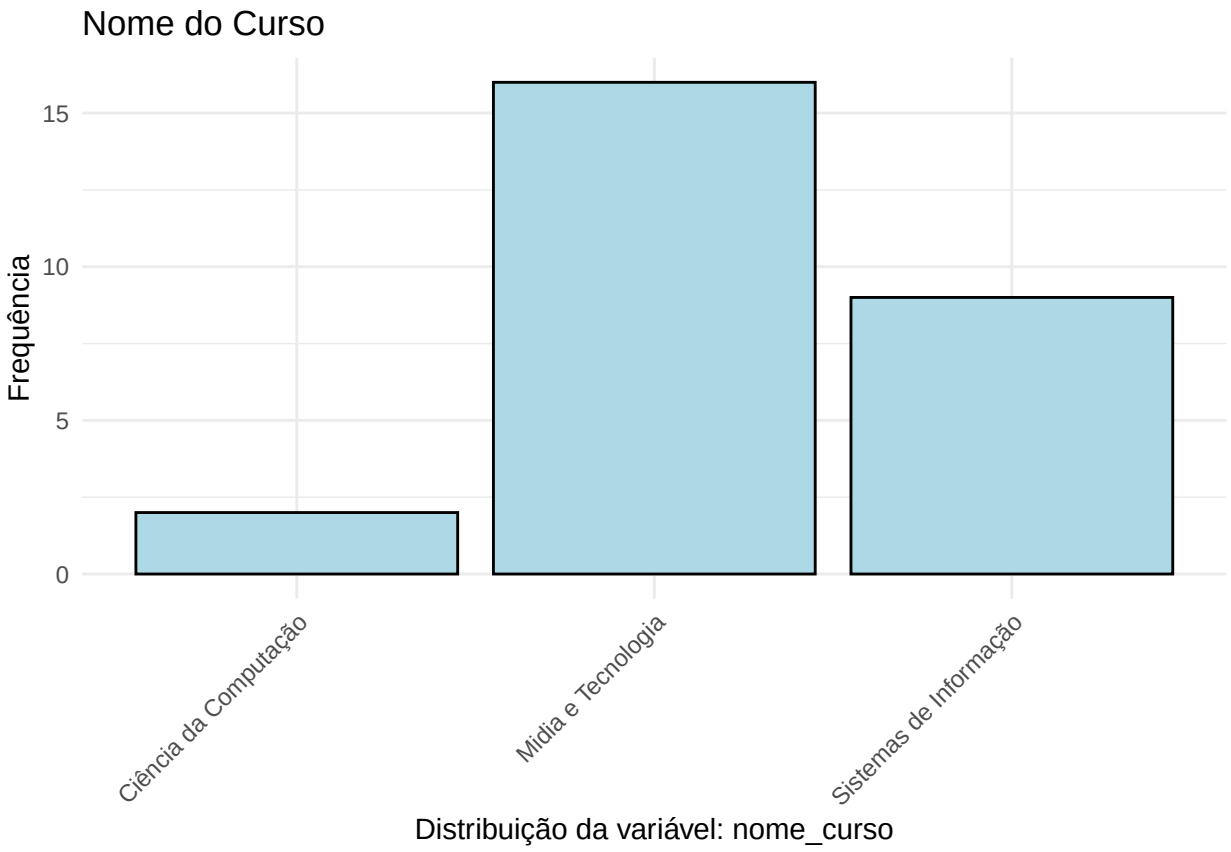
```
         y = "Frequência") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotacionar os rótulos do eixo X

  # Exibir o gráfico
  print(p)

  indice = indice + 1
}
```

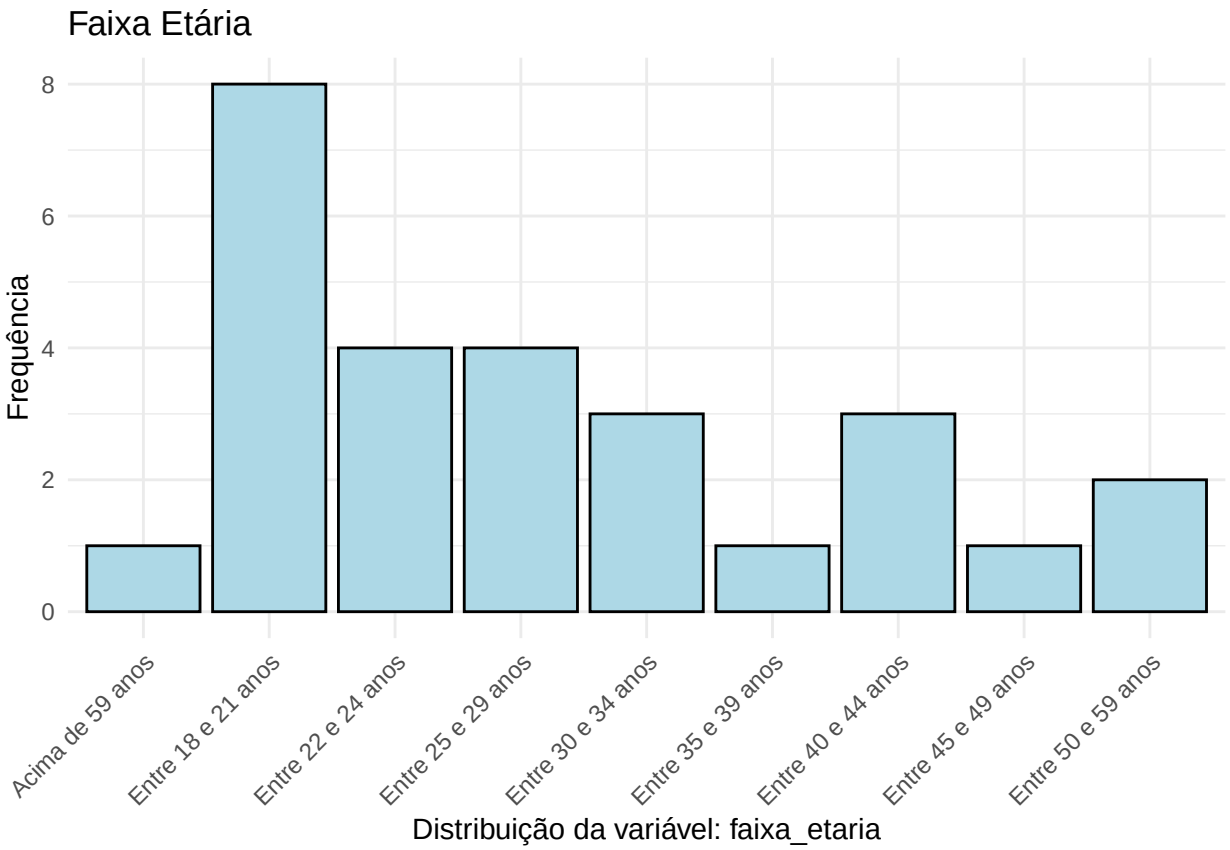


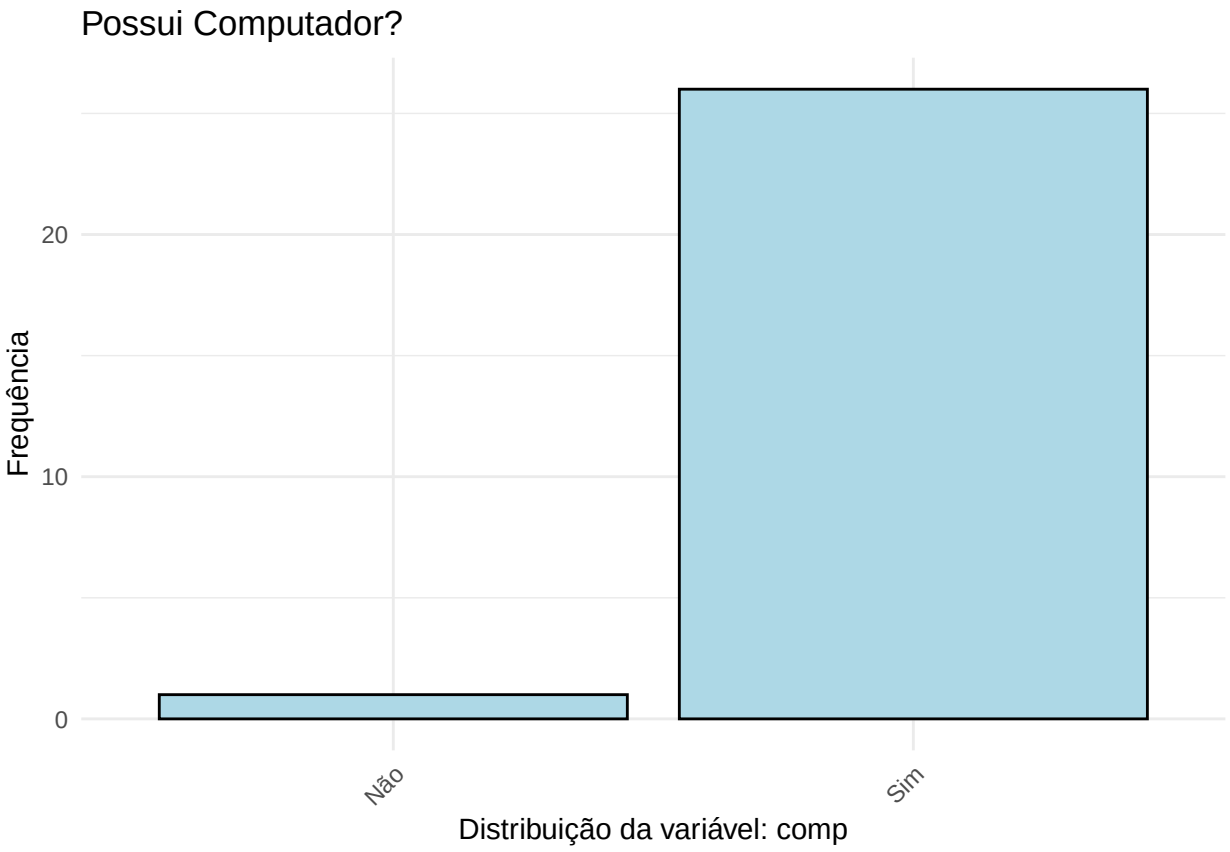

Tipo de Aluno
Frequência
15
10
5
0
Graduação
Outro
Pós−graduação
Distribuição da variável: tipo_alunoa

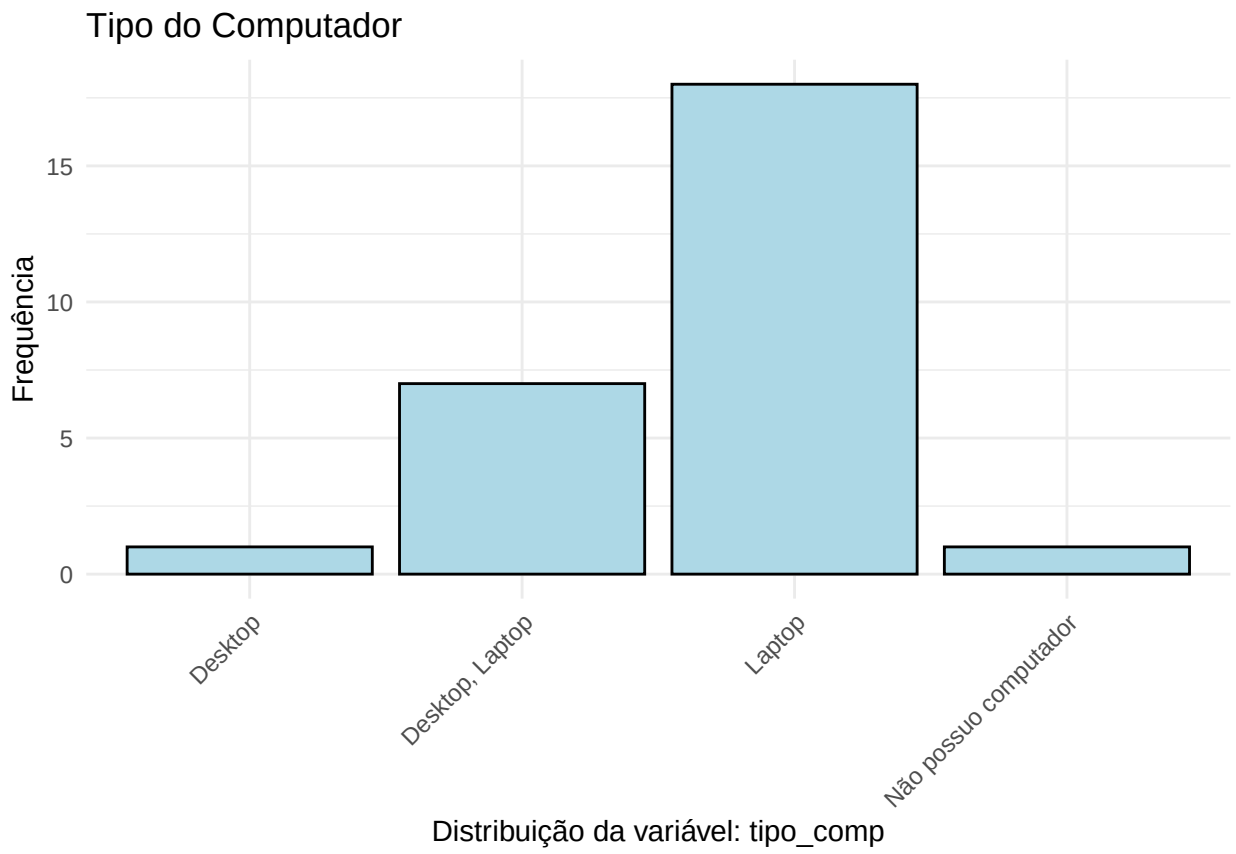


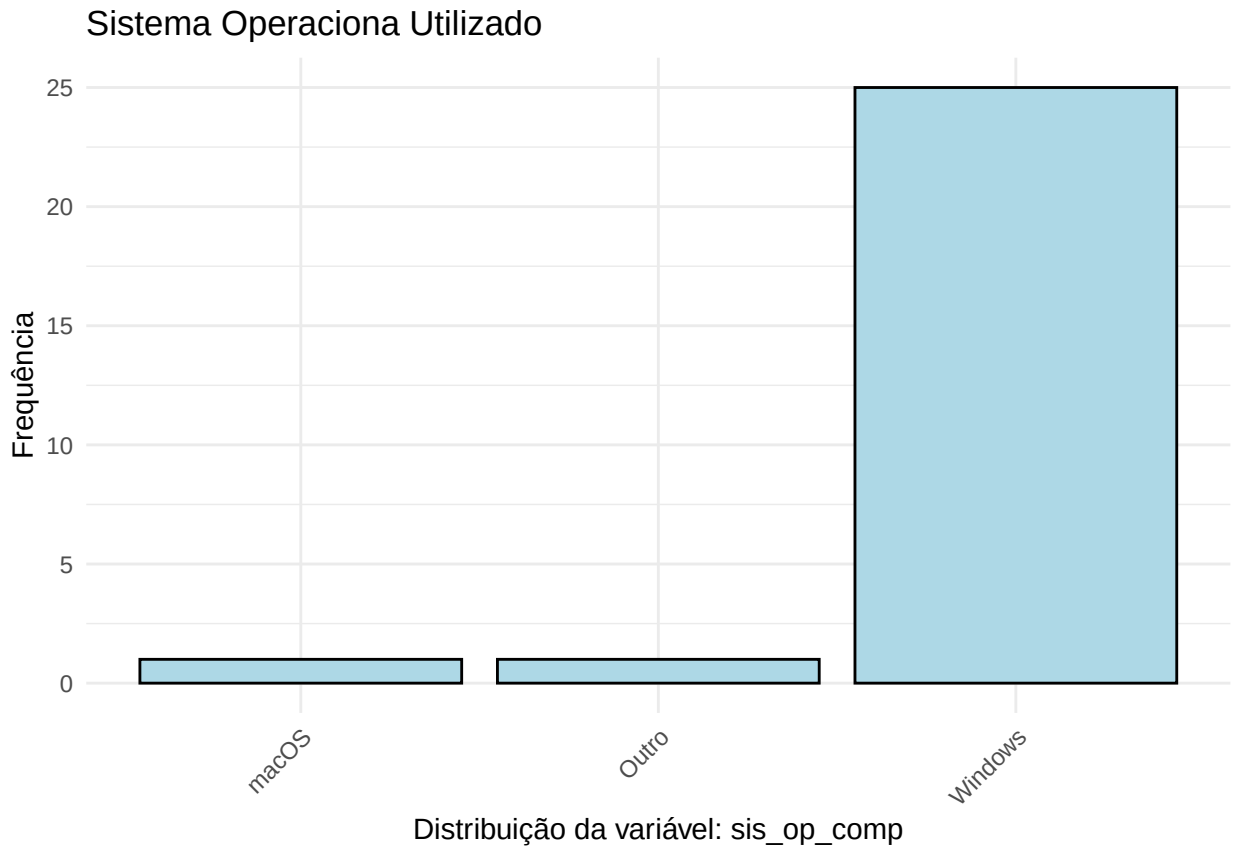
Distribuição da variável: formacao

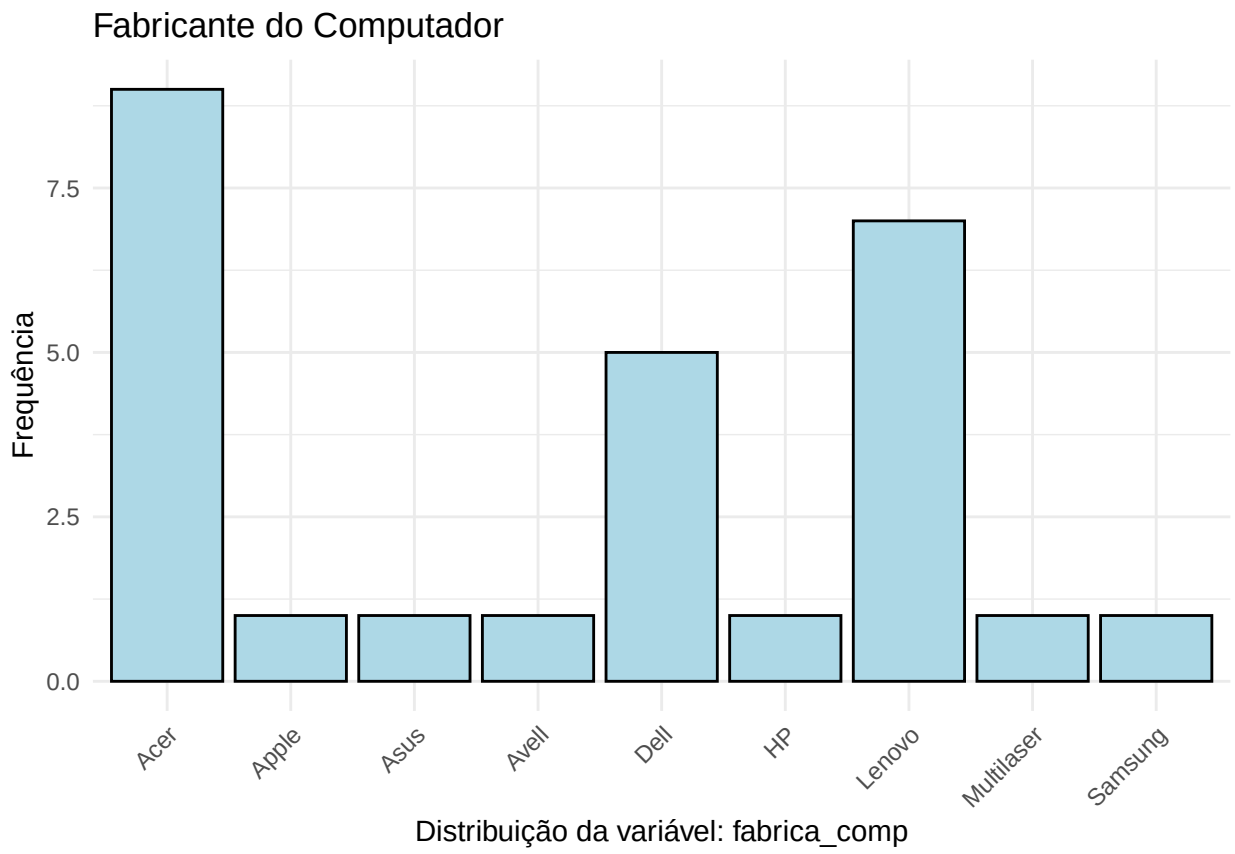

Nome do Curso
Frequência
15
10
5
0
Ciência da Computação
Mídia e Tecnologia
Sistemas de Informação
Distribuição da variável: nome_curso

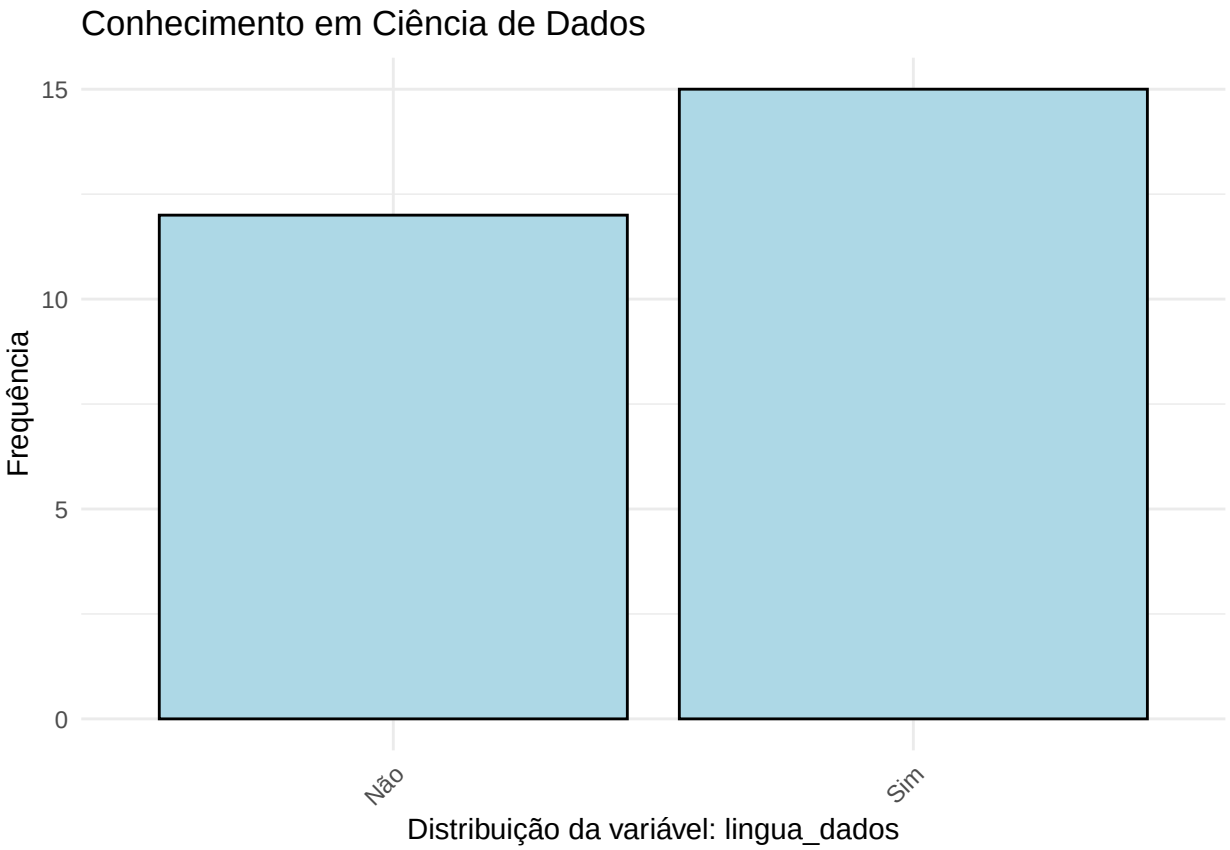

Faixa Etária
Frequência
8
6
4
2
0
Acima de 59 anos
Entre 18 e 21 anos
Entre 22 e 24 anos
Entre 25 e 29 anos
Entre 30 e 34 anos
Entre 35 e 39 anos
Entre 40 e 44 anos
Entre 45 e 49 anos
Entre 50 e 59 anos
Distribuição da variável: faixa_etaria

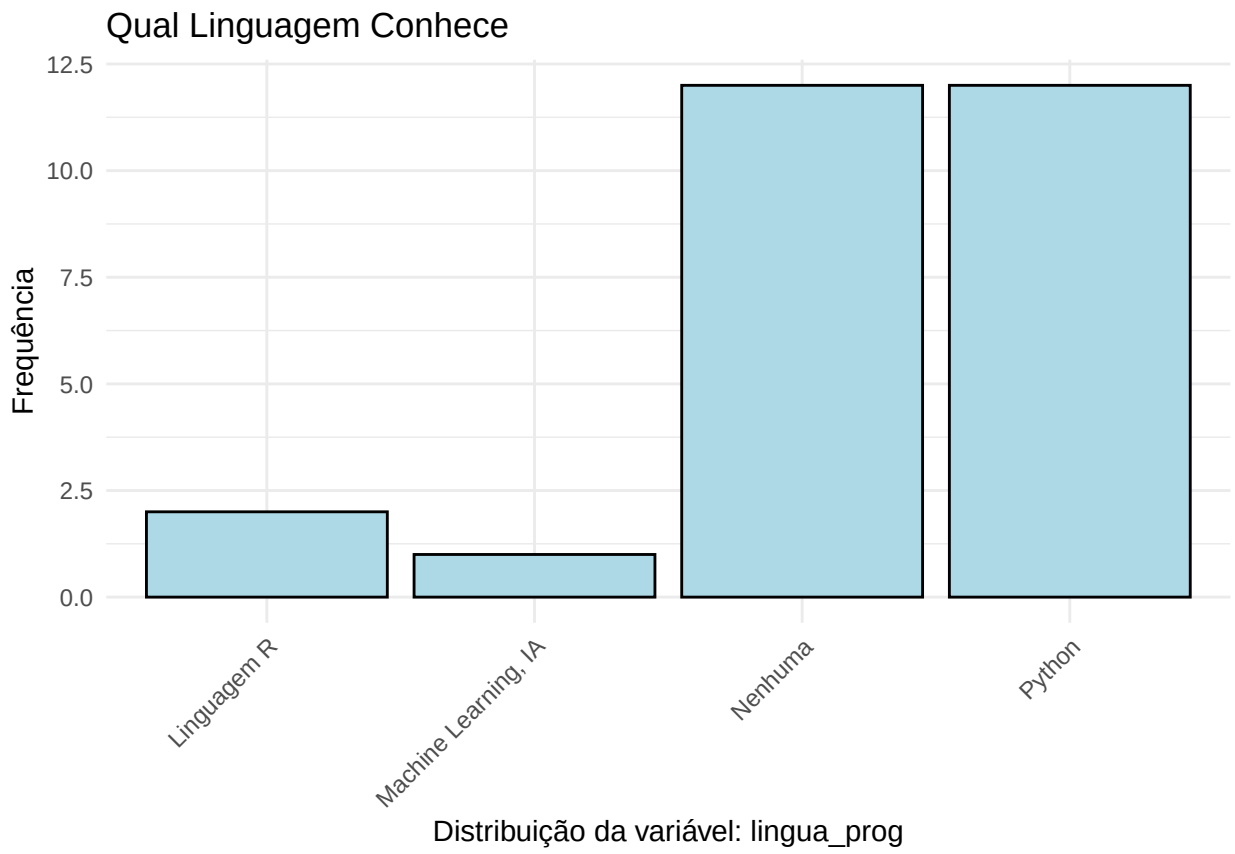

Possui Computador?
Frequência
20
10
0
Não
Sim
Distribuição da variável: comp


Tipo do Computador
Frequência
15
10
5
0
Desktop
Desktop, Laptop
Laptop
Não possuo computador
Distribuição da variável: tipo_comp


Sistema Operaciona Utilizado
Frequência
25
20
15
10
5
0
macOS
Outro
Windows
Distribuição da variável: sis_op_comp

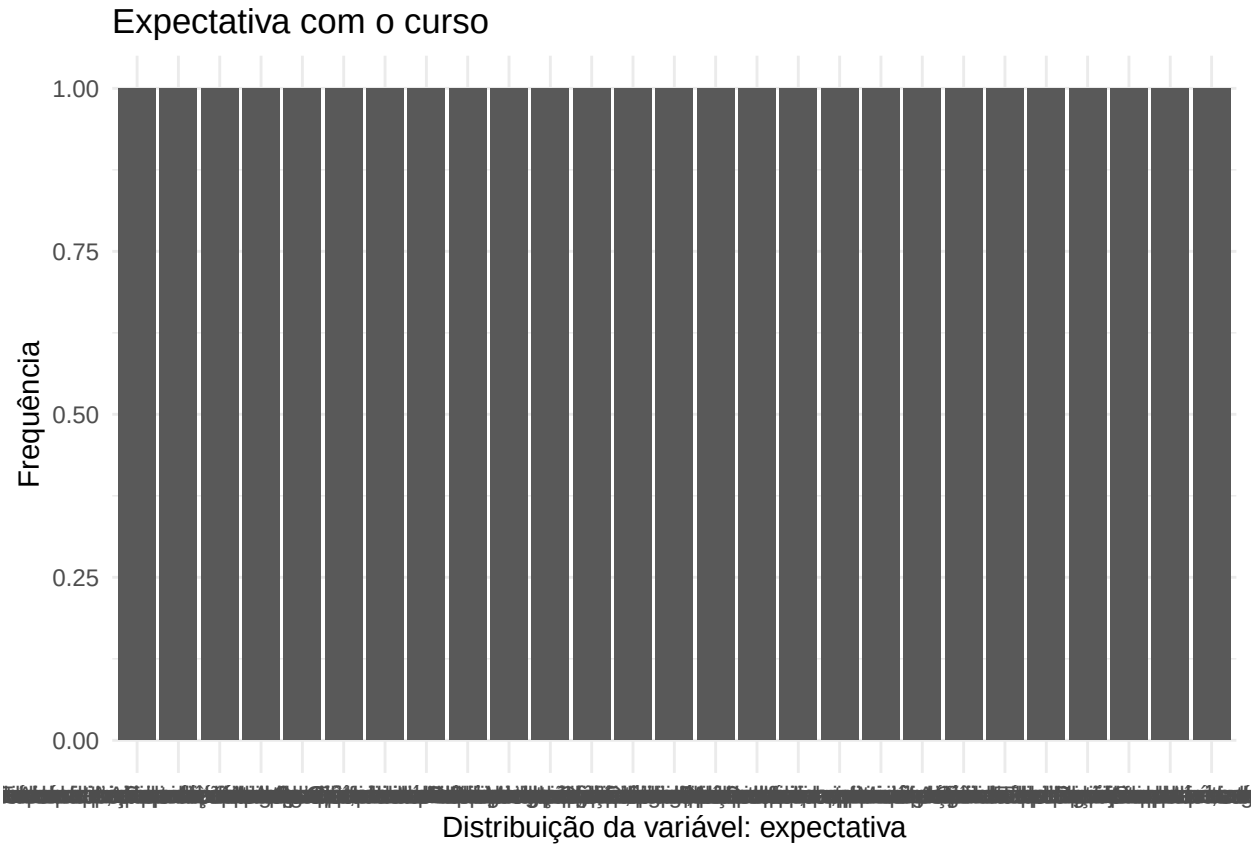

Fabricante do Computador
Frequência
7.5
5.0
2.5
0.0
Acer
Apple
Asus
Avell
Dell
HP
Lenovo
Multilaser
Samsung
Distribuição da variável: fabrica_comp


Conhecimento em Ciência de Dados
15
10
5
0
Frequência
Não
Sim
Distribuição da variável: lingua_dados



A visualização dos dados de varíaveis do tipo *texto* podem ficar confusas quando visualizadas em gráficos do tipo pizza ou barras, por exemplo, como pode ser visualizado na execução do bloco de código a seguir:

```
ggplot(dados, aes(x = expectativa)) +
         geom_bar() +
         labs(title =  "Expectativa com o curso",
          x = paste("Distribuição da variável:", "expectativa"),
          y = "Frequência") +
          theme_minimal()
```



Como observado na plotagem do gráfico anterior da variável `expectativa` se torna difícil de compreender o que está sendo mostrado. Algumas das razões para isso são: muitas categorias diferentes; categorias com nomes longos; rótulos ilegíveis e gráfico visualmente complexo e difícil de interpretar.

Por essas razões, outras formas de visualização podem ser mais eficazes para variáveis de texto como o recurso de **nuvem de palavras** ou **word cloud** que melhoram a compreensão visual do conteúdo da variável.

Uma *Word Cloud* é uma representação visual de um conjunto de palavras, onde o tamanho de cada palavra é proporcional à sua frequência de ocorrência em um determinado texto. Essa técnica é amplamente utilizada para identificar os termos mais relevantes e frequentes em um texto, permitindo uma análise rápida e intuitiva das informações contidas nele (TECHNER, 2023).

Esse recurso em como objetivo principal fornecer uma visualização rápida e intuitiva da distribuição das palavras em um texto. Ao observá-la, é possível identificar os termos que aparecem com mais frequência e, portanto, têm maior destaque na representação gráfica (FREITAS, 2023).

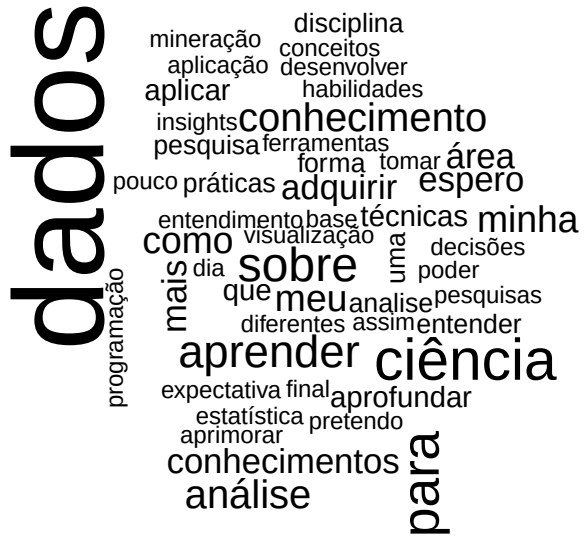
A representação gráfica da nuvem de palavras facilita a identificação do que é mais relevante e sua importante no contexto em questão. Geralmente, os vocábulos mais frequentes são exibidas com maior tamanho ou cor mais intensa, enquanto as menos frequentes são apresentadas com tamanho reduzido ou cor mais clara, afirma Freitas (2023).

Essa visualização auxilia na identificação de temas principais, tópicos de destaque e palavras-chave, permitindo uma compreensão mais rápida e abrangente do conteúdo (TECHNER, 2023).

Um exemplo simples de visualização pode ser observada com a execução do bloco de código a seguit:

```
# Visualizar "expectativa" como nuvem de palavras - versão simplificada
# Carregar pacotes necessários
library(wordcloud)
```

```
library(RColorBrewer)
wordcloud(words = dados$expectativa, min.freq = 2)
```



Como se pode observar na execução do trecho de código anterior, a visualização de dados na forma de word cloud (nuvem de palavras) pode ser especialmente eficaz para variáveis de texto por várias razões, tais como:

1. **Facilidade de Identificação**:
   - **Palavras Frequentes**: As palavras mais frequentes aparecem maiores, facilitando a identificação rápida dos termos mais comuns.
   - **Visualmente Atraente**: A disposição aleatória e o uso de diferentes tamanhos e cores tornam a visualização mais atraente e fácil de interpretar.
2. **Resumo Visual**:
   - **Visão Geral**: Oferece uma visão geral rápida do conteúdo textual, destacando as palavras-chave sem a necessidade de ler todo o texto.
   - **Destaque de Padrões**: Ajuda a identificar padrões e tendências nas palavras usadas.
3. **Engajamento**:
   - **Interatividade**: Muitas ferramentas permitem a criação de word clouds interativas, onde os usuários podem clicar nas palavras para obter mais informações.
   - **Apelo Estético**: A natureza visualmente agradável das word clouds pode tornar apresentações e relatórios mais envolventes.

4. **Simplicidade**:

    - **Fácil de Criar**: Com bibliotecas como **wordcloud** em R, é simples gerar uma word cloud a partir de um conjunto de dados.
    - **Intuitivo**: Mesmo pessoas sem conhecimento técnico podem entender rapidamente a informação apresentada.

Para realizar a interpretação das características de uma variável texto, pode-se utilizar a *técnica de análise de conteúdo*, uma forma de interpretar dados textuais de maneira sistemática e objetiva e envolve a codificação de texto em categorias que representam conceitos ou temas (MORETTI, 2021 e ORTEGA, 2024).

Para realizar uma análise de conteúdo da variável **expectativa** na linguagem R, pode-se aproveitar a funcionalidade de processamento e análise de texto oferecida pelo pacote **quanteda** (https://quanteda.io/).

O **quanteda** é um pacote em R projetado para a análise quantitativa de dados textuais. O pacote foi criado por Kenneth Benoit e Kohei Watanabe e é mantido pela Quanteda Initiative CIC (https://quanteda.org/), uma organização sem fins lucrativos voltada para a promoção de software de análise de texto de código aberto.

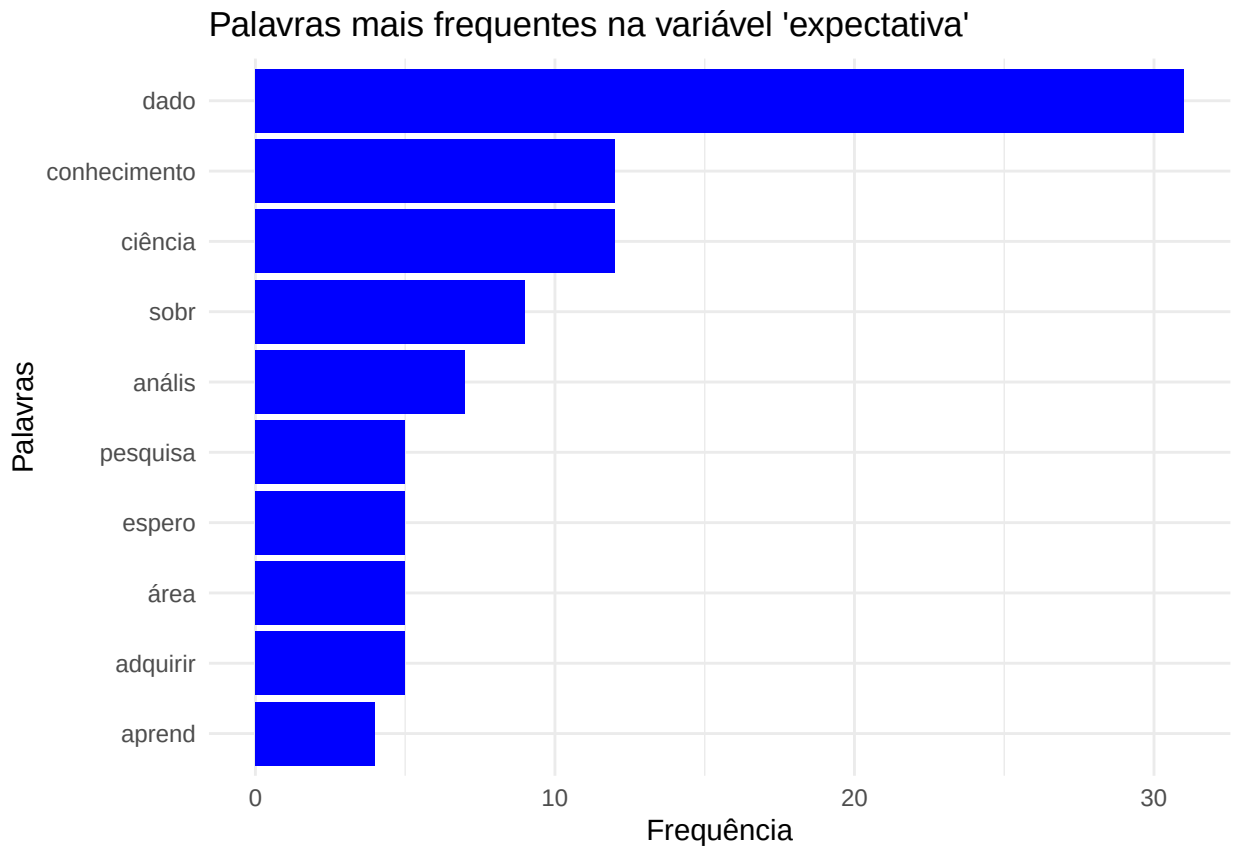
O pacote **quanteda** oferece uma ampla gama de funções para manipulação e análise de texto, incluindo tokenização, criação de matrizes de documentos e termos (DTM) e visualizações como nuvem de palavras. Além disso, é especialmente eficiente para a análise de texto em larga escala e tarefas de NLP (Processamento de Linguagem Natural).

O trecho de código a seguir, realiza uma análise de conteúdo da variável expectativa utilizando o pacote quanteda:

```r
# Carregar pacotes necessários
library(quanteda)
library(quanteda.textstats)
library(quanteda.textplots)

# Selecionar a variável expectativa
text_data <- dados$expectativa

# Remover NAs e entradas vazias
text_data <- text_data[!is.na(text_data) & text_data != ""]

# Criar um corpus com quanteda
corpus_expectativa <- corpus(text_data)

# Tokenizar o texto (convertendo para palavras)
tokens_expectativa <- tokens(corpus_expectativa,
                             remove_punct = TRUE,
                             remove_numbers = TRUE) %>%
    tokens_remove(stopwords("pt")) %>% # Remover stopwords em português
    tokens_wordstem() # Aplicar stemming (reduzir palavras à sua raiz)

# Criar uma matriz de documentos e termos (Document-Feature Matrix - DFM)
dfm_expectativa <- dfm(tokens_expectativa)

# Exibir as palavras mais frequentes
top_terms <- textstat_frequency(dfm_expectativa, n = 10)
print(top_terms)
```

```
##          feature frequency rank docfreq group
## 1           dado        31    1      22   all
## 2       ciência        12    2      12   all
## 3  conhecimento        12    2      11   all
## 4           sobr         9    4       9   all
## 5         anális         7    5       5   all
## 6         espero         5    6       5   all
## 7           área         5    6       5   all
## 8       pesquisa         5    6       5   all
## 9       adquirir         5    6       5   all
## 10        aprend         4   10       4   all
```

```
# Plotar as 10 palavras mais frequentes
ggplot(top_terms, aes(x = reorder(feature, frequency), y = frequency)) +
    geom_bar(stat = "identity", fill = "blue") +
    coord_flip() +
    labs(title = "Palavras mais frequentes na variável 'expectativa'",
         x = "Palavras",
         y = "Frequência") +
    theme_minimal()
```



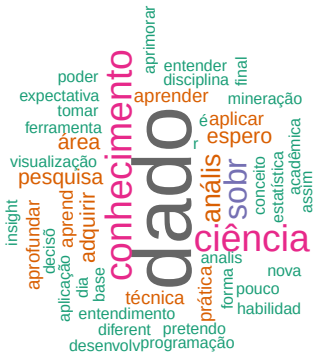

```
# Executar a análise de frequência das palavras
print("Análise de frequência das palavras")
```

```
## [1] "Análise de frequência das palavras"
```

```r
topfeatures(dfm_expectativa)
```

```
##         dado      ciência conhecimento         sobr       anális       espero
##           31           12           12            9            7            5
##         área     pesquisa     adquirir       aprend
##            5            5            5            4
```

```r
# Gerar uma nuvem de palavras
textplot_wordcloud(dfm_expectativa, max_words = 100,
                   min.freq = 2, random.order = FALSE,
                   rot.per = 0.35, colors = brewer.pal(8, "Dark2"))
```



Uma breve explicação das etapas desenvolvidas no bloco de código anterior:

1. **Carregar os Pacotes**
   - quanteda: Principal pacote para processamento e análise de texto.
   - quanteda.textstats: Para gerar estatísticas textuais como a frequência de termos.
   - quanteda.textplots: Para gerar visualizações, como nuvens de palavras
2. **Criação de um Corpus**
   - O texto contido na variável `expectativa` é convertido em um corpus usando corpus() do quanteda. Um corpus é a estrutura usada para manipulação e análise de grandes volumes de texto.

3. **Tokenização**
   - O corpus é tokenizado com a função tokens(), que quebra o texto em palavras.
   - **Remoção de Stopwords**: As stopwords em português são removidas usando tokens_remove().
   - **Stemming**: Aplicamos tokens_wordstem() para reduzir as palavras às suas raízes (por exemplo, "trabalhando" e "trabalho" se tornam "trabalh").
4. **Matriz de Documentos e Termos (DFM)**
   - A DFM (Document-Feature Matrix) é criada com dfm(). Cada documento (resposta) é representado por uma linha, e cada coluna representa um termo (palavra).
5. **Análise de Frequência**
   - As 10 palavras mais frequentes são extraídas usando textstat_frequency().
   - Um gráfico de barras com as palavras mais frequentes é criado com ggplot2.
6. **Nuvem de Palavras**
   - A nuvem de palavras é gerada com textplot_wordcloud(), exibindo as 100 palavras mais frequentes.

Como observado na nuvem de palavras anterior, as word clouds são ótimas ferramentas de comunicação e visualização. No entanto, elas também têm seus limites e compreender suas limitações é fundamental para se saber quando usá-las e, portanto, também quando não usá-las.

De acordo com Van den Rul (2019), nuvens de palavras é essencialmente uma ferramenta *descritiva*. Portanto, deve ser utilizada apenas para capturar **insights qualitativos básicos**. Visualmente atraentes, é uma ótima ferramenta para **iniciar** um debate, uma apresentação ou uma análise. No entanto, sua análise é limitada a insights que simplesmente não têm o mesmo calibre que uma análise estatística mais extensa.

4. Insights e Conclusões

A análise exploratória realizada fornece insights iniciais sobre a distribuição das variáveis, bem como uma visualização textual das expectativas expressas pelos respondentes.

5. Documentação utilizando o Quarto

Para encerrar todo o processo de análise exploratória, é gerado (publicado) um documento em Quarto que contém todo o código R, as visualizações (gráficos e word cloud) as análises e conclusões.
Ao final do processo, o documento é publicado como um Relatório. Para realizar tal processo, deve-se seguir os seguintes passos:

1. Salvar o código acima em um arquivo com a extensão **.qmd** (por exemplo, seu-perfil-v1.qmd).
2. Renderizar o documento com o Quarto clicando o botão **Render** no RStudio.
3. Visualizar o Relatório: O comando Render gera um arquivo HTML que se pode abrir no navegador para visualizar o relatório completo.

Referências

DAMIANI, Athos et. al. Ciência de Dados em R – Curso-R**.** 12 jul. 2022. Disponível em: https://livro.curso-r.com/3-13-outros-tópicos.html#fatores. Acesso em: 03 set. 2024.

LAURETTO, Marcelo**.** Capítulo 1: Análise Exploratória de Dados, Estatística Computacional-SIN5008, EACH-USP, São Paulo. 2011. Disponível em: http://www.each.usp.br/lauretto/SIN5008_2011/aula01/

aula1#:~:text=Etapas%20da%20AED&text=realizar%20um%20exame%20gráfico%20das,de%20dados%20ausentes%20(missing)%3B. Acesso em: 003 set. 2024.

FREIRE, Sergio Miranda. Introdução ao R. Ed. do Autor. 2021. Disponível em: https://www.lampada.uerj.br/arquivosdb/_book2/sessao.html. Acesso em: 03 set. 2024.

FREITAS, Camila. Nuvem de palavras: o que é e como utilizar para melhorar a escrita. Blog Guia da Carreira, Quero Educação. 18 jul. 2023. Disponível em: https://www.guiadacarreira.com.br/blog/nuvem-de-palavras. Acesso em: 03 set. 2024.

GUERRA, Saulo et. al. Ciência de Dados com R – Introdução. IBPAD: Instituto Brasileiro de Pesquisa e Análise de Dados. 18 nov. 2020. Disponível em: https://cdr.ibpad.com.br. Acesso em: 03 set. 2024.

MANGINI, Florencia. Best practices organizing data science projects. Blog Florencia Mangini. 9 dez. 2028. Disponível em: https://www.thinkingondata.com/how-to-organize-data-science-projects/. Acesso em: 11 set. 2024.

MORETTI, Isabela. O que é análise de conteúdo? Veja o passo a passo do método. Blog ViaCarreira. 11 jan. 2011. Disponível em: https://viacarreira.com/analise-de-conteudo/. Acesso em: 12 set. 2024.

MULDOON, Ariel. Analysis essentials: An example directory structure for an analysis using R, Blog Very Staticious, 29 out. 2018. Disponível em: https://aosmith.rbind.io/2018/10/29/an-example-directory-structure/. Acesso em: 11 set. 2024.

OLIVEIRA, Mateus A. de. Funções de descrição in:R - Aula 3, estatística descritiva e gráficos. 14 jul. 2021. Disponível em: https://rstudio-pubs-static.s3.amazonaws.com/790368_5929534db9cc4d45848cfcb1e0ce0047.html#:~:text=A%20função%20mais%20famosa%20para,medida%20de%20disperção%20dos%20dados. Acesso em: 03 set. 2024.

ORTEGA, Cristina. Análise de conteúdo: o que é e como funciona em estudos qualitativos. Blog do Software de pesquisa QuestionPro. 2024. Disponível em: https://www.questionpro.com/blog/pt-br/analise-de-conteudo/. Acesso em 11 set. 2024.

ROGONODO, Harefa. Data Science Project Folder Structure. Github. 2023. Disponível em: https://github.com/hardefarogonondo/data-science-project-folder-structure. Acesso em: 11 set. 2024.

RODRIGUES, Gregório Victor. Valores missing: O que são? Onde vivem? Do que se alimentam? From Patients to Python. 28 mai. 2023. Disponível em: https://www.linkedin.com/pulse/valores-missing-o-que-são-onde-vivem-do-se-alimentam-rodrigues/. Acesso em: 03 set. 2024.

SARDAR, Abha Singh. Demystifying Data Visualization: A Deep Dive into "A Layered Grammar of Graphics". Blog Medium. 12 fev. 2024. disponível em: https://medium.com/@abhapratiti27/demystifying-data-visualization-a-deep-dive-into-a-layered-grammar-of-graphics-e387751be870#:~:text=At%20the%20heart%20of%20Wickham Acesso em: 11 set. 2024.

SOUZA, Ricardo Lehtonen R. Capítulo 4 Gráficos in: Análises de Dados com R. Editora do Autor. 11 ago. 2023. Disponível em: https://bookdown.org/ricardo_lehtonen/anlise_de_dados_com_r/gr%C3%A1ficos.html. Acesso em: 11 set. 2024.

SUBEX. Corpus. Blog Artificial Intelligence. 30 set. 2023. Disponível em: https://www.subex.com/article/corpus/#:~:text=A%20corpus%20can%20be%20made,AI%20and%20machine%20learning%20systems. Acesso em: 03 set. 2024.

TATSCH, Jônatan. Data Frames (Quadro de Dados) in: Análise de Dados Ambientais com R. Disciplina Análise de Dados Ambientais com R, curso de Graduação em Meteorologia, Departamento de Física da Universidade Federal de Santa Maria. 29 jan. 2021. Disponível em: https://lhmet.github.io/adar-ebook/data-frames-quadro-de-dados.html. Acesso em: 03 set. 2024.

TECHNER. O que é Word Cloud? Blog Techner. 02 dez. 2023. Disponível em: https://techner.com.br/glossario/o-que-e-word-cloud/. Acesso em: 03 set. 2024.

Van den Rul, Céline. How to Generate Word Clouds in R: Simple Steps on How and When to Use Them. Towards Data Science. Blog Medium. 15 out. 2019. Disponível em: https://towardsdatascience.com/create-a-word-cloud-with-r-bde3e7422e8a. Acesso em: 03 set. 2024.

WICKHAM, Hadley. A Layered Grammar of Graphics. Journal of Computational and Graphical Statistics, 19(1), 3–28. 2010. Disponível em: https://doi.org/10.1198/jcgs.2009.07098.

WICKHAM, Hadley. ggplot2: Elegant Graphics for Data Analysis. Springer, 2016.Acesso em: 11 set. 2024.

WILKINSON, Leland. The Grammar of Graphics. Springer, 1999.